



UNIVERSIDAD CENTRAL DEL ECUADOR

FACULTAD DE INGENIERÍA CIENCIAS FÍSICAS Y MATEMÁTICA
CARRERA DE INGENIERÍA INFORMÁTICA.

“SISTEMA WEB PARA INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN
DEL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI”

TRABAJO DE GRADUACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO INFORMÁTICO.

AUTOR:

Suntaxi Gualotuña Luz Lorena

TUTOR:

Ing. César Morales Mejía, Msc.

QUITO – ECUADOR

2014



DEDICATORIA

A DIOS, quién supo guiarme por el buen camino, por ser mi fortaleza en los momentos de debilidad, por permitirme culminar con éxito el esfuerzo de todos estos años de estudio.

A mi padre, por ser un gran hombre, por los ejemplos de perseverancia y constancia que lo caracterizan, gracias por estar conmigo siempre, por su apoyo, comprensión, y por ayudarme en los recursos económicos.

A mi madre, por quererme mucho, creer en mí y porque siempre me apoyo, gracias por tu paciencia, tus consejos, valores, y sobre todo por decirme esas palabras sabias que siempre tiene para mis enojos, mis tristezas y mis alegrías, y más que todo por ser mi amiga. Todo lo que soy te lo debo a ti.

A mis hermanos, por ser parte importante de mi vida, por compartir conmigo momentos de éxitos y fracasos, y apoyarme siempre.

A mi novio, por sus consejos por su amor y apoyo incondicional durante mi etapa estudiantil y ahora profesional,

A mis sobrinas ya que con sus risas llenan mi vida de alegría.



AGRADECIMIENTO

La realización de esta tesis fue posible gracias a la intervención brindada por mi tutor Ing. César Morales Mejía Msc., por su ayuda constante, su orientación metodológica, y por su continuo estímulo para que este proyecto de tesis sea una realidad.

Al Ing. René Carrillo e Ing. Mauro Rosas, quienes con la mejor predisposición permitieron la finalización del mismo.

Quiero expresar mi agradecimiento y compromiso con el Patronato de Promoción Social de Rumiñahui especialmente con Abg. Andrés Medina Coordinador General, ya que me facilitó la información y los recursos necesarios para la realización y desarrollo de este proyecto de tesis.

Un agradecimiento especial a la gloriosa Universidad Central del Ecuador, ya que fue un excelente espacio de formación y estudio, y en especial por todo lo que me ha brindado personal y profesionalmente.

Y a todas aquellas personas que en forma directa e indirectamente contribuyeron a que este trabajo de tesis pudiera llevarse a cabo.



AUTORIZACION DE LA AUTORIA INTELECTUAL

Yo, SUNTAXI GUALOTUÑA LUZ LORENA, en calidad de autor de la tesis realizada sobre **“SISTEMA WEB PARA INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN DEL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI”**, por la presente autorizo a la UNIVERSIDAD CENTRAL DEL ECUADOR, hacer uso de todos los casos contenidos que me pertenecen o de parte de los que contienen esta obra, con fines estrictamente académicos o de investigación.

Los derechos que como autor me corresponden, con excepción de la presente autorización, seguirán vigentes a mi favor, de conformidad con lo establecido en los artículos 5, 6, 8, 19 y demás pendientes de la ley de la Propiedad Intelectual y su Reglamento.

Quito, 28 de enero del 2014.

Suntaxi Gualotuña Luz Lorena

C.I. 1716199904



CERTIFICACIÓN

En calidad de Tutor del proyecto de Investigación “**SISTEMA WEB PARA INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN DEL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI**”, presentado y desarrollado por la señorita: LUZ LORENA SUNTAXI GUALOTUÑA, previo a la obtención del Título de Ingeniero Informático, considero que el proyecto reúne los requisitos necesarios.

En la Ciudad de Quito a los 28 días del mes de Enero del 2014.

Ing. César Augusto Morales Mejía Msc.

CATEDRÁTICO

CARRERA INGENIERÍA INFORMÁTICA



APROBACIÓN DEL DIRECTOR DE TESIS



FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICAS
CARRERA DE INGENIERÍA INFORMÁTICA

Quito D.M., 28 de noviembre del 2013

Señor Matemático

Juan Pablo Cevallos

DIRECTOR DE LA CARRERA DE INGENIERÍA INFORMÁTICA

Presente.-

Señor Director:

Yo, Ingeniero César Augusto Morales Mejía, docente de la Carrera de Ingeniería Informática de la Facultad de Ingeniería Ciencias Físicas y Matemática de la Universidad Central del Ecuador.

Certificó:

Luego de las revisiones técnicas realizadas por mi persona al desarrollo del proyecto de investigación “SISTEMA WEB PARA INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN DEL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI”, llevado a cabo por parte de la egresada de la carrera de Ingeniería Informática, señorita, Luz Lorena Suntaxi Gualotuña, con C.I.1716199904 ha concluido, consecuentemente el indicado estudiante podrá continuar con los tramites de graduación correspondientes de acuerdo a lo que estipula las normativas y disposiciones legales.

Por la atención que digno al presente, reitero mi agradecimiento.

Atentamente

Ing. César Augusto Morales Mejía Msc.

CATEDRÁTICO

CARRERA INGENIERÍA INFORMATICA



OFICIO

		UNIVERSIDAD CENTRAL DEL ECUADOR FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y MATEMÁTICA CARRERA DE INGENIERÍA INFORMÁTICA	
DIRECCIÓN		Teléfonos: 2542026 - 2236987 ext. 218 Fax 2236039	
		Oficio N° 0023-2014-DC-UMC Quito, D.M., 4 de febrero del 2014	
Señores Ing. René Carrillo Ing. Mauro Rosas PROFESORES DE LA CARRERA DE INGENIERÍA INFORMÁTICA Presente.			
Señores Profesores:			
A fin de dar cumplimiento a lo dispuesto en el "Reglamento para la Obtención de los Títulos Profesionales en la Facultad de Ingeniería, Ciencias Físicas y Matemática", aprobado por el H. Consejo Universitario, en sesión del 31 de octubre de 2011; agradeceré a usted, calificar el Trabajo de Graduación titulado: "SISTEMA WEB DE INSCRIPCIÓN PARA EL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI" realizado por la estudiante: LUZ LORENA SUNTAXI GUALOTUÑA , requisito previo a la obtención del título de INGENIERA INFORMÁTICA , en base al Formulario del Resultado del Trabajo de Graduación , que me permito remitirle.			
Este formulario, deberá enviarse a la Secretaría General de la Facultad en un plazo no mayor a ocho días .			
Atentamente, Mat. Juan Pablo Cevallos DIRECTOR, CARRERA INGENIERÍA INFORMÁTICA			
RECIBI CONFORME		FIRMA	
Ing. René Carrillo			
Ing. Mauro Rosas			
Sol. No. C-052			
Buenos días			



RESULTADO DEL TRABAJO DE GRADUACIÓN



DIRECCIÓN

UNIVERSIDAD CENTRAL DEL ECUADOR
FACULTAD DE INGENIERÍA, CIENCIAS FÍSICAS Y
MATEMÁTICA
CARRERA DE INGENIERÍA INFORMÁTICA

Teléfonos: 2542006 - 2236987 ext. 218 Fax 2236039

RESULTADO DEL TRABAJO DE GRADUACIÓN

CARRERA DE: INGENIERÍA INFORMÁTICA

Quito, ... 17/01/2014

Señorita: LUZ LORENA SUNTAXI GUALOTUÑA

TEMA: "SISTEMA WEB DE INSCRIPCIÓN PARA EL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI"

CALIFICACIÓN:

TRIBUNAL	PROFESOR (A)	NOTA SOBRE VEINTE		FIRMA
		NUMERO	LETRAS	
PROFESOR TITULAR	Ing. René Carrillo	20	VEINTE	
PROFESOR TITULAR	Ing. Mauro Rosas	19	Diecinueve	
PROMEDIO		19.5	Diecinueve y medio	

Dra. Katheryne Carrión Valdivieso
SECRETARIA ABOGADA



CONTENIDO

DEDICATORIA	ii
AGRADECIMIENTO	iii
AUTORIZACION DE LA AUTORIA INTELECTUAL.....	iv
CERTIFICACIÓN	v
APROBACIÓN DEL DIRECTOR DE TESIS	vi
OFICIO	vii
RESULTADO DEL TRABAJO DE GRADUACIÓN	viii
CONTENIDO	ix
LISTA DE TABLAS	xii
LISTA DE FIGURAS.....	xiii
RESUMEN	xiv
ABSTRACT.....	xv
CERTIFICACIÓN DE TRADUCCIÓN.....	xvi
TÍTULO DEL TRADUCTOR.....	xvii
INTRODUCCIÓN	xviii
CAPÍTULO 1	1
1. PRESENTACIÓN DEL PROBLEMA	1
1.1. Planteamiento del Problema.....	1
1.2. Formulación del Problema	2
1.3. Interrogantes de la Investigación	3
1.4. Objetivos de la Investigación	4
1.5. Objetivo General	4
1.6. Objetivos Específicos.....	4
1.7. Justificación	5
CAPÍTULO 2.....	6
2. REVISIÓN BIBLIOGRÁFICA	6
2.1. Antecedentes	6
2.2. Fundamentación Teórico.....	7
2.2.1. Análisis y diseño	7
2.2.2. Implementación.....	7
2.2.3. Seguimiento	7
2.2.4. Evaluación.....	7



2.2.5.	Control	8
2.2.6.	Modelo de desarrollo	9
2.2.6.1.	Modelo Espiral	9
2.2.7.	Diagramas UML	12
2.2.8.	Software libre.....	19
2.2.9.	Lenguaje de Programación Java	20
2.2.9.1.	Características del lenguaje Java.....	22
2.2.10.	IDE para Programación en Java: Eclipse	24
2.2.11.	Aplicaciones web con JPA, EJB, JSF y PrimeFaces	24
2.2.12.	Servidor de Aplicaciones Jboss.....	32
2.3.	Identificación de las Variables	34
2.3.1.	Variables independientes.....	35
2.3.2.	Variables dependientes.....	35
2.4.	Hipótesis	36
CAPÍTULO 3.....		37
3.	METODOLOGÍA	37
3.1.	Diseño de la Investigación	37
3.1.1.	Metodología RUP/ Easy.....	38
3.1.2.	Ventajas de la metodología RUP/Easy.....	39
3.2.	Metodología de desarrollo	39
3.2.1.	Modelo espiral	39
3.2.1.1.	Fase de ingeniería	41
3.2.1.2.	Fase de elaboración	43
3.2.1.3.	Fase de producción.....	44
3.3.	Arquitectura MVC	48
3.3.1.	Capa modelo	49
3.3.2.	Capa controlador	49
3.3.3.	Capa vista.....	49
3.3.4.	Tecnología WEB	49
3.3.5.	¿Por qué utilizar MVC?.....	50
3.3.6.	Ventajas de MVC	50
CAPÍTULO 4.....		51
4.	DIAGRAMACIÓN DEL SISTEMA	51
4.2.	Diagramas de casos de uso	52



4.2.1.	Caso de uso: Ingreso al sistema	53
4.2.2.	Caso de uso: Administrador-Usuario-Creación de usuarios.....	54
4.2.3.	Caso de uso: Modificar alumno	55
4.2.4.	Caso de uso: Inscripción alumno	57
4.2.5.	Caso de uso: Matriculación alumno.....	58
4.3.	Diagramas de Secuencia	60
CAPÍTULO 5		61
5. INSTALACIÓN DE HERRAMIENTAS		61
5.1. Herramientas para el desarrollo.		62
5.1.1 Instalar MySQL Community Server 5.6.12.....		62
5.1.1.1 Crear base de datos patronatobd en MySQL Server 5.6.....		70
5.1.2. Instalación de jboss-6.1.0.Final en CentOS 6		72
CAPÍTULO 6.....		73
6. CONCLUSIONES Y RECOMENDACIONES.....		73
6.1. Conclusiones		73
6.2. Recomendaciones.....		74
BIBLIOGRAFÍA		76
GLOSARIO		77
ANEXOS		81



LISTA DE TABLAS

Tabla 4.1. Actores y acciones principales.....	53
Tabla 4.2. Inventario de casos de uso.	53
Tabla 4.3. Caso de uso CU-001	54
Tabla 4.4. Caso de uso CU-003.....	55
Tabla 4.5. Caso de uso CU-004.....	56
Tabla 4.6. Caso de uso CU-005.....	57
Tabla 4.7. Caso de uso CU-006.....	59



LISTA DE FIGURAS

Figura 2.1. Modelo Espiral	10
Figura 2.2. Modelo EspiralWinWin.....	10
Figura 2.3. Modelo Espiral seis regiones	11
Figura 2.4. Usuario	15
Figura 2.5. Caso de uso.....	15
Figura 2.6. Asociación	16
Figura 2.7. Dependencia	16
Figura 2.8. Generalización	16
Figura 2.9. Ejemplo de diagrama de caso de uso.....	17
Figura 2.10. Ejemplo de diagrama de secuencia.....	18
Figura 2.11. Objeto	18
Figura 2.12. Diagrama eje vertical.....	18
Figura 2.13. Mensaje al mismo objeto	19
Figura 2.14. Mensaje de un objeto a otro objeto.....	19
Figura 2.15.Duke, la mascota de Java.....	22
Figura 2.16.. Tecnologías Java EE para el desarrollo de aplicaciones web	25
Figura 2.17.Estándar de los contenedores Java EE.....	26
Figura 2.18. Servicios proporcionados por los contenedores	27
Figura 2.19 .Definición de atributos y relaciones de una Entidad.	30
Figura 2.20. Triangulo del Proyecto.....	33
Figura 3.1. Arquitectura MVC	48
Figura 4.1. Modelado del negocio	51
Figura 4.2. Ingreso al sistema.....	53
Figura 4.3. Creación de usuarios	54
Figura 4.4. Modificar alumno.....	55
Figura 4.5. Inscripción alumno	57
Figura 4.6. Matriculación alumno	58
Figura 4.7. Diagrama de secuencia Sistema de Inscripción.....	60



RESUMEN

SISTEMA WEB DE INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN DEL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI.

Este desarrollo provee al Patronato de Promoción Social de Rumiñahui una herramienta informática, que pretende solucionar sus problemas de almacenamiento de información y su posterior tratamiento para la generación de reportes al automatizar sus procesos de inscripción de los alumnos a los cursos de capacitación. La herramienta desarrollada es sencilla y amigable, de tal forma que resulta de fácil manipulación para cualquier persona. A demás está desarrollada bajo un ambiente web lo que permite que sea accedida desde cualquier lugar en donde se tenga una conexión a internet. Permite varios niveles de seguridad al asignar perfiles a los usuarios que intervienen en cada proceso desarrollado. Por otra parte los reportes son generados fácilmente y ayudan a obtener resultados estadísticos en cuanto a alumnos, capacitadores y eventos.

DESCRIPTORES:

SISTEMA WEB DE INSCRIPCIÓN/ CURSOS DE CAPACITACIÓN
SOCIAL/PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI
/SOFTWARE LIBRE



ABSTRACT

WEB REGISTRATION SYSTEM TO TRAINING COURSES TO THE BOARD OF SOCIAL PROMOTION OF RUMIÑAHUI.

This development provides the Board of Social Promotion Rumiñahui a software tool that aims to solve their problems of information storage and subsequent treatment for generating reports to automate your processes student enrollment training courses. The tool developed is a simple and user-friendly, such that manipulation is easy for anyone. In addition it is developed under a web environment that allows it to be accessed from any location where an Internet connection have. The tool allows multiple security levels to assign profiles to users involved in each process developed. Moreover reports are easily generated and help obtain statistical results regarding students, trainers and events.

DESCRIPTORS:

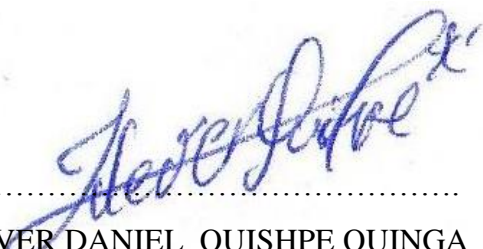
WEB REGISTRATION SYSTEM / SOCIAL TRAINING COURSES / SOCIAL PROMOTION BOARD OF RUMIÑAHUI / FREE SOFTWARE



CERTIFICACIÓN DE TRADUCCIÓN

En calidad de Traductor del proyecto de investigación “SISTEMA WEB PARA INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN DEL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI”, presentado y desarrollado por LUZ LORENA SUNTAXI GUALOTUÑA, se certifica la traducción, previo a la obtención del Título de Ingeniero Informático.

En la ciudad de Quito, a los 28 días del mes de Enero del 2014



.....

KLEVER DANIEL QUISHPE QUINGA
TRADUCTOR



TÍTULO DEL TRADUCTOR





INTRODUCCIÓN

Con el transcurso del tiempo la tecnología avanza, los centros educativos se sienten en la necesidad de adquirir tecnología para el mejoramiento de sus procesos, con el fin de garantizar un eficaz funcionamiento. Es importante confrontar el desarrollo en el campo de la informática, tal vez la palabra clave es la comunicación, pues el hombre ha sido capaz de comunicarse más fácilmente, las computadoras liberan al hombre de las abrumadoras tareas de efectuar rutinas masivas y le permite emplear su inteligencia en tareas más estimulantes e interesantes.

La investigación realizada permitió desarrollar un estudio sobre la necesidad de un Sistema de Inscripción para los cursos de Capacitación del Patronato de Promoción Social de Rumiñahui y las herramientas necesarias para solucionar los problemas más comunes de los procesos de inscripción manual.

El sistema de inscripción consiste en ingresar en el sistema con un usuario y clave, de acuerdo a los siguientes procesos.

Proceso de administración.- Para realizar cambios en el sistema, el administrador será el encargado, el cual conoce cómo funciona el sistema.

Proceso de inscripción y matriculación.- El estudiante será inscrito previo deberá registrarse, siguiente podrá inscribirse a un curso. El proceso de inscripción lo podrá realizar el alumno o el administrador del sistema, para la matriculación deberá confirmar el pago y se procede a actualizar su estado cambiándolo de inscrito a matriculado.

Reportes.- El sistema deberá entregar los siguientes reportes: listado de alumnos por curso y capacitadores, matrículas de los estudiantes semanal.

El proyecto se fundamenta en la utilización de una metodología que garantice que el producto final a entregar cumpla con las expectativas generadas alrededor de esta nueva herramienta y para evitar costos de licenciamiento se ejecutó el desarrollo con la utilización de software libre¹.

¹(Recuperado el 01 de diciembre del 2012) de http://www.smaldone.com.ar/opinion/docs/sl_informatica.html



CAPÍTULO 1

En éste capítulo se presenta una breve descripción de la motivación del presente trabajo de investigación, dando a conocer el problema planteado junto con la solución para hacer frente al problema. Adicionalmente se detalla los objetivos, contenido y justificación del proyecto.

1. PRESENTACIÓN DEL PROBLEMA

1.1. Planteamiento del Problema

El Patronato de Promoción Social de Rumiñahui, es una institución creada con la finalidad de dar servicio a la población, que cumple con proyectos de carácter social y de atención a los sectores más vulnerables y necesitados del Cantón, como base para la investigación el proceso de inscripción manual, la cual tiene como objetivo determinar mediante la evaluación de profesionales en la materia que tanta eficacia, confiabilidad, veracidad y rapidez tendría el control automatizado de INSCRIPCIÓN es.

El principal objetivo de esta investigación es realizar un estudio sobre las causas que originan la necesidad de llevar un sistema de inscripción en el Patronato de Promoción Social de Rumiñahui y las herramientas necesarias para erradicar los problemas más comunes en cuanto a los procesos de inscripción manual.

En este sentido se entiende como sistema de inscripción, el procedimiento a seguir al inicio de cada periodo de curso de capacitación, el sistema de inscripción es manual y a partir de observaciones y de levantamiento de información que arrojo la investigación se pudo detectar que la mayoría de los procedimientos tienen inconvenientes, entre ellos se pueden destacar: pérdida o duplicación de información debido a la transcripción manual o forma de almacenamiento.

Por lo tanto, para liberar este proceso de la abrumadora tarea de efectuar INSCRIPCIÓN es manuales que ocasiona congestionamiento, crecimiento de la documentación se ofrece una herramienta de trabajo que consiste en un sistema de información más exacta y rápida para en un corto período de tiempo dar inicio a las actividades de manera eficiente e integra, además proveer información actualizada



tanto de alumnos, profesores, eventos y certificados para el administrador para la toma de decisiones, facilitar información para el departamento contable y ofrecer reportes de acuerdo a la necesidad del usuario.

Con todo lo planteado anteriormente, lo que quiero es dar servicio a la Comunidad por parte de la Universidad Central del Ecuador, dando así un apoyo al Programa de Capacitación del Patronato de Promoción Social de Rumiñahui quien tiene como objetivo brindar a la Comunidad, un espacio de desarrollo de conocimientos y aprendizaje, generando alternativas reales para la creación de microempresas que permitan mejorar los ingresos del núcleo familiar.

1.2. Formulación del Problema

Actualmente el Patronato de Promoción Social de Rumiñahui no posee un sistema que automatice sus tareas académicas administrativas, lo que ocasiona que cada periodo de capacitación la planificación académica se vuelva una tarea sumamente compleja, que frecuentemente ocasiona gran pérdida de tiempo y esfuerzo, iniciando con la inscripción de alumnos en cada periodo, seguido con la dificultad que involucra manejar un amplio número de registros en hojas de cálculo, y junto a la generación de reportes que ocupan una gran cantidad de recursos materiales y humanos, que podrían ser canalizados en otras actividades.

Los procesos automatizados, son factores de vital importancia en la organización, en esta era se ha observado una gran innovación y desarrollo de tecnologías de la información que ha permitido la evolución de computadores que son capaces de producir sistemas de información a la sociedad con aportes a los distintos sectores económicos, sociales, políticos, educativos, entre otros.

Es necesario que la información sea procesada y almacenada de una forma más efectiva para agilizar los procesos de inscripción y así lograr un control integral de las actividades de inscripción. Con el desarrollo de un sistema automatizado que abarque las necesidades y una mayor relación con los requerimientos del personal, proporcionará una mejor efectividad en el manejo del flujo y procesamiento de los datos.

Así al establecer este sistema poseerá un impacto psicosocial positivo en la sociedad,



la cual proporcionará, una información confiable, agilizando y facilitando el trabajo en el proceso de inscripción a los cursos de capacitación del Patronato de Promoción Social de Rumiñahui. Cabe destacar que se ha conseguido un total apoyo a la idea de implantar un sistema automatizado de inscripción mediante la web, además con la implementación del sistema informático de INSCRIPCIÓN es a los cursos de capacitación se va a mejorar de manera sustancial la entrega de certificados, mayor información de los cursos en cuanto a costos y horarios, obteniendo satisfacción de los alumnos, capacitadores, departamento contable y administrador.

Todo el personal está involucrando en este proyecto porque quieren hacer realidad un proyecto que traería un soporte técnico confiable y acorde a los avances de la era tecnológica de la información.

1.3. Interrogantes de la Investigación

El presente tema de tesis nace de la observación de los problemas que se suscitan en el Patronato de Promoción Social para la Inscripción a los cursos de capacitación, limitando la productividad del usuario, ya que aún se realiza la manipulación de la documentación lo que no permite la gestión de conocimiento.

De acuerdo a la investigación se desea contestar las siguientes interrogantes:

- ✓ ¿De qué manera ayuda al Programa de Capacitación el Sistema Web Para Inscripción a los Cursos de Capacitación del Patronato de Promoción Social?
- ✓ ¿Qué técnicas y métodos se utilizarán en el desarrollo del sistema para la obtención de conocimiento?
- ✓ ¿Cuáles herramientas de software libre se adaptan mejor a las necesidades del Patronato de Promoción Social del Cantón Rumiñahui?
- ✓ ¿Cómo se lleva a cabo el proceso actual de inscripción y matriculación en el Patronato de Promoción Social del Cantón Rumiñahui?
- ✓ ¿Cuáles son los pasos que definen los procesos de inscripción y matriculación de un estudiante?
- ✓ ¿Cuáles serán las ventajas y desventajas que traería la automatización del Sistema de inscripción y matriculación en el Patronato de Promoción Social?



- ✓ ¿Qué recursos serán necesarios para la implementación del sistema que lleva el proceso de inscripción?
- ✓ ¿Cuál será el mecanismo y política de seguridad más adecuada que se utilizará en el desarrollo del sistema?
- ✓ ¿Qué estrategia se utilizará para adaptar el personal encargado del manejo del sistema?

1.4. Objetivos de la Investigación

Construir una solución informática, cuyo diseño se base en software libre y que cumpliendo con altos estándares de calidad aporte con cualidades que apoyen los principales procesos que se realizan en el Patronato de Promoción Social de Rumiñahui en el Programa de Capacitación, para la inscripción y matriculación y así apoyar la gestión que realiza esta entidad pública.

1.5. Objetivo General

- ✓ Desarrollar un sistema web software para la automatización del proceso de inscripción de los cursos de capacitación del Patronato de Promoción Social de Rumiñahui.

1.6. Objetivos Específicos

- ✓ Implementar un Sistema Web que permita mejorar el proceso de inscripción a los cursos de capacitación en el Patronato de Promoción Social de Rumiñahui utilizando herramientas de distribución libre.
- ✓ Analizar los requerimientos para la automatización de los procesos de inscripción y matriculación.
- ✓ Determinar las ventajas y desventajas que traería para el Patronato de Promoción Social de Rumiñahui automatizando los formatos manuales de inscripción y matriculación.



- ✓ Precisar los recursos que son necesarios para implementar un sistema automatizado que lleve a cabo el control del proceso de inscripción.
- ✓ Implementar niveles de seguridad para garantizar que la navegación, y/o ejecución de tareas sean acciones basadas en roles y perfiles de usuarios dentro del aplicativo.
- ✓ Diseñar una interfaz con el cual los usuarios puedan introducir los datos de manera eficaz.
- ✓ Almacenar la información que se maneja para obtener disponibilidad de la información y tener una organización cero papeles.
- ✓ Facilitar el proceso contable.

1.7. Justificación

El análisis detallado, estará sustentado en diagramas UML de casos de uso y de secuencia, permitirá contestar las interrogantes planteadas para esta investigación, así determinando los principales procesos realizados por los cursos de capacitación del Patronato de Promoción Social de Rumiñahui y por quienes los ejecutan para generar perfiles de usuario que ayuden a organizar a los actores de diferentes procesos y brindar ciertos niveles de seguridad para el proyecto.

Siendo la Universidad Central del Ecuador una de las más prestigiosas universidades en busca de acreditación “A”, nosotros queremos brindar servicios y soluciones reales al problema del proceso de inscripción a los cursos de capacitación del Patronato de Promoción Social de Rumiñahui.

Así al establecer este sistema poseerá un impacto psicosocial positivo en la sociedad, la cual proporcionará a la misma, una información confiable, agilizando y facilitando el trabajo en el proceso de inscripción y matriculación. Se ha conseguido un total apoyo a la idea de implantar un sistema automatizado de inscripción, el personal que labora en esta entidad se siente partidario por hacer realidad un proyecto que traería un soporte técnico confiable y acorde a los avances de la era tecnológica de la información..



CAPÍTULO 2

En este capítulo se presenta un estudio de las diversas herramientas y tecnologías para el desarrollo de las aplicaciones web robusta, potente, de alta disponibilidad y que simplifica enormemente su desarrollo.

2. REVISIÓN BIBLIOGRÁFICA

2.1. Antecedentes

Entre los antecedentes relacionados con el tema están un grupo de estudios, los cuales son una necesaria referencia ya que orientan en el desarrollo del mismo.

Carrero (2002), "Aplicación de Tecnología Web en los Sistemas de Información. Caso: Sistema de Control de Calificaciones". En su investigación presenta como objetivo crear los cimientos de un sistema de base de datos con las notas de los alumnos a través de una página dinámica en la Web, en donde los docentes, los alumnos y los representantes serán los principales usuarios. En sus conclusiones señala: La meta final será llegar a crear una comunidad virtual alrededor del sitio Web, con Chat directamente con los profesores, y correo electrónico, a través de la cual primeramente se minimice el tiempo de respuesta a la solicitud de información y en segundo lugar, que esta información sea válida y confiable.

El aporte que esta tesis dará a la investigación es que el autor planteó la creación de una base de datos con las notas de los alumnos, como un proyecto factible, es decir, desarrolló un sistema de información que permite llevar el control de las calificaciones de una forma rápida y confiable, en el trabajo en estudio igualmente, se piensa desarrollar una aplicación tecnológica, con el fin de automatizar un proceso manual de inscripción.

Este desarrollo informático por tratarse de un proyecto de tesis, debe evitar costes por compras de licencias de software, así también debe cumplir con los requerimientos del Coordinador del Patronato de Promoción Social de Rumiñahui.

Una vez implantado y en funcionamiento el sistema, y con el apoyo de las autoridades del Patronato de Promoción Social de Rumiñahui se puede implantar adicionalmente cursos virtuales, ventas de libros.



2.2. Fundamentación Teórico

En el presente proyecto para su desarrollo se plantea un marco teórico base ya que el sistema de inscripción y matriculación considera las características esenciales en el equipo o sistema, las cuales son el presupuesto máximo autorizado, el tiempo máximo de entrega del equipo requerido, la compatibilidad con el lenguaje computacional actual, y las características mínimas requeridas de rendimiento de computadoras.

Los puntos establecidos se basan en la identificación del desarrollo de software.

2.2.1. Análisis y diseño

Es la disposición y orden de las partes dentro de un todo. También puede entenderse como un sistema de conceptos coherentes enlazados, cuyo objetivo es precisar la esencia del objeto de estudio.

Tanto la realidad como el lenguaje tienen estructura. Uno de los objetivos de la semántica y de la ciencia consiste en que la estructura del lenguaje refleje fielmente la estructura de la realidad.

2.2.2. Implementación

Es la programación de un determinado algoritmo² en un lenguaje específico. Por ejemplo, un algoritmo en pseudocódigo³ se implementa en forma de código de un lenguaje de programación.

2.2.3. Seguimiento

Se define como un informe continuo, realizado por una tercera parte independiente, sobre la situación y las cuestiones vinculadas a los riesgos identificados en la implementación del proyecto.

2.2.4. Evaluación

Tiene como finalidad determinar el grado de eficacia y eficiencia, con que han sido empleados los recursos destinados a alcanzar los objetivos previstos, posibilitando la

² (Recuperado el 01 de diciembre del 2012) de <http://www.alegsa.com.ar/Dic/algoritmo.php>

³ (Recuperado el 01 de diciembre del 2012) de http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n



determinación de las desviaciones y la adopción de medidas correctivas que garanticen el cumplimiento adecuado de las metas presupuestadas.

2.2.5. Control

Es una etapa primordial en la administración pues, aunque una empresa cuente con magníficos planes, una estructura organizacional adecuada y una dirección eficiente, el ejecutivo no podrá verificar cuál es la situación real de la organización si no existe un mecanismo que se cerciore e informe si los hechos van de acuerdo con los objetivos.

El Análisis, Diseño e Implementación del sistema de inscripción y matriculación implementada para el Patronato de Promoción Social de Rumiñahui depende de la conciliación entre una formulación adecuada y completa del módulo, una planificación operacional y realista, métodos de seguimiento, evaluación y control.

Se llama módulo a una parte de un sistema, un proceso dinámico de retro-alimentación permanente en cada una de las fases; por ejemplo, la definición de la situación inicial (uno de los pasos de la planificación), nos da elementos para revisar y ajustar la situación final deseada que hemos definido previamente (en uno de los primeros pasos de la planificación); cuando realizamos el análisis, diseño, éste nos aporta información que nos ayuda en caso necesario para corregir las diferentes necesidades, porque nos permite saber si lo que estamos haciendo nos lleva o no a alcanzar la situación final y los resultados que nos hemos propuesto. Igual ocurre con el control, que además de permitirnos adecuar el rumbo (cuando se realiza dentro del trayecto de intervención), nos facilita aprender de la experiencia, compartir con otros esos aprendizajes y a partir de ellos, generar nuevos conocimientos.

Análisis, diseño e implementación del sistema de inscripción y matriculación es en términos generales, un conjunto de procedimientos y mecanismos de recolección y análisis de información sobre:

- ✓ Reducir el tiempo de respuesta.
- ✓ Reducir errores y mejorar la entrada de datos.
- ✓ Reducir costos mediante la eliminación de duplicados innecesarios.
- ✓ Agilizar consultas sobre la base de reportes precisos.



- ✓ Reducir el tiempo de procesamiento de datos.
- ✓ Disponer de un único dispositivo capaz de localizar datos de alumnos, profesores, certificados, cursos y costos.

El sistema permite, por tanto, contar con información relevante y oportuna para la toma de decisiones en cuanto a las mejores estrategias posibles para alcanzar lo que nos proponemos (la planificación), realizar los reajustes y/o modificaciones necesarios considerándolos cambios que se van dando en el contexto y en la situación de los grupos beneficiarios; y la forma en que vamos avanzando hacia el logro de los resultados esperados.

2.2.6. Modelo de desarrollo

Un modelo de desarrollo es una representación abstracta de un proceso de software, cada modelo representa el proceso de desarrollo de software de una manera en particular. A pesar de estar definidos claramente, no representan necesariamente la realidad de cómo se debe desarrollar el software, sino que establece un enfoque común. Un modelo puede ser modificado y adaptado de acuerdo a las necesidades del software en desarrollo, también denominado ciclo de vida⁴ del desarrollo de software. Hay varios modelos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso.

Para realizar nuestra aplicación se utilizará el “Modelo Espiral”

2.2.6.1. Modelo Espiral

Es un modelo de proceso de software evolutivo donde se conjuga la naturaleza de construcción de prototipos. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software que no se basa en fases claramente definidas y separadas para crear un sistema.

⁴(Recuperado el 01 de diciembre del 2012) de <http://www.ojovisual.net/galofarino/modeloespiral.pdf>



Figura 2.1. Modelo Espiral

Fuente: (Recuperado 01 de diciembre del 2012) <http://modeloespiral.blogspot.com/>

En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones la versión incremental podría ser un modelo en papel o un prototipo, durante las últimas iteraciones se producen versiones cada vez más completas del sistema diseñado.

Modelo Espiral WINWIN



Figura 2.2. Modelo EspiralWinWin

Fuente: (Recuperado 01 de diciembre del 2012) <http://modeloespiral.blogspot.com/>

El modelo en espiral de seis regiones puede adaptarse y aplicarse a lo largo de la vida del software de computadora. Una visión alternativa del modelo en espiral puede ser considerada examinando el eje de punto de entrada en el proyecto.



Figura 2.3. Modelo Espiral seis regiones

Fuente: (Recuperado 01 de diciembre del 2012) <http://modeloespiral.blogspot.com/>

Las regiones de tareas que componen este modelo son:

- ✓ Comunicación con el cliente: las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- ✓ Planificación: las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos.
- ✓ Análisis de riesgos: las tareas requeridas para evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto.
- ✓ Ingeniería: las tareas requeridas para construir una o más representaciones de la aplicación.
- ✓ Construcción y adaptación: las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- ✓ Evaluación del cliente: las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementación durante la etapa de instalación.

Ventajas Modelo Espiral

- ✓ Puede adaptarse y aplicarse a lo largo de la vida del software de computadora.



- ✓ Como el software evoluciona a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.
- ✓ El modelo en espiral permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.
- ✓ El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y si se aplica adecuadamente debe reducir los riesgos antes de que se conviertan en problemas.
- ✓ En la utilización de grandes sistemas a doblado la productividad.

Desventajas Modelo Espiral

- ✓ Resulta difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.
- ✓ Debido a su elevada complejidad no se aconseja utilizarlo en pequeños sistemas.
- ✓ Genera mucho tiempo en el desarrollo del sistema.
- ✓ Modelo costoso.
- ✓ Requiere experiencia en la identificación de riesgos.

2.2.7. Diagramas UML

El proceso mediante el cual se define el alcance del proyecto que se espera satisfaga las necesidades del cliente, se ejecutará realizando varias reuniones con el Coordinador del Patronato de Promoción Social del Cantón Rumiñahui Sr. Andrés Medina involucrado directamente con los procesos que se pretenden automatizar. En estas reuniones se levantará la información necesaria para identificar los casos de uso que cada actor requiere dentro de las acciones que éste realiza. La información recopilada será traducida a diagramas UML (Unified Modeling Language) que es un estándar para el modelado de sistemas de software.



UML es el lenguaje de modelado de sistemas de software, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Podemos organizar nuestro conocimiento utilizando tres coordenadas:

- ✓ **Espacio:** Existe objetos condicionados por una estructura que son capaces de colaborar para cumplir una misión.
- ✓ **Tiempo:** Ocurren eventos dentro de una secuencia acotada por escenarios (Flujos de trabajo).
- ✓ **Información:** Los Objetos y Eventos producen hechos relevantes para un Actor, registrables y comunicables con una determinada escala de abstracción y granularidad.

UML es una notación visual orientada a la elaboración de modelos de procesos y/o productos. Dispone de un repertorio limitado de unidades con significado (clases, acciones, objetos, estados, casos de uso), y una gramática que define un conjunto de reglas de combinación para formar otras unidades de significado más complejo.

Es mucho más eficiente visualizar la complejidad de una organización en un reducido número de diagramas que en un centenar de páginas. También es mucho más ágil mantener los diagramas actualizados con los cambios de necesidades y nuevos enfoques de la organización.

Con un número reducido de elementos UML y sus reglas de combinación, es posible construir y comunicar estructuras y funcionalidad muy compleja.

Se usa UML cuando necesitamos:

- ✓ Definir un problema que afecta a una organización (análisis).
- ✓ Plantear una solución de diseño (abstracción).
- ✓ Modelar procesos de negocio (optimización de flujos de trabajo).
- ✓ Construir un producto de software (concreción de una abstracción).
- ✓ Certificar la coherencia, completitud y usabilidad del producto (calidad).
- ✓ Evaluar la arquitectura de una organización (conocimiento).



2.2.7.1. Diagramas de casos de uso

Un diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Son los principales medios para capturar la funcionalidad del software a implementar, representan la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa, describen acciones y reacciones al comportamiento de un sistema desde el punto de vista del usuario.

La especificación formal de un caso de uso incluye:

Requerimientos. Es un contrato que el caso de uso realizará alguna acción y proveerá algún valor al sistema.

Restricciones. Son reglas formales y limitaciones bajo la cual opera un caso de uso e incluye condiciones pre, post e invariantes. Una pre-condición especifica lo que ya debe de haber ocurrido. Una post-condición documenta que será cierto una vez que el caso de uso está completo. Una invariante especifica que será cierto durante el tiempo que el caso de uso opera.

Escenarios. Los escenarios son descripciones formales del flujo de eventos que ocurren durante la instancia de un Caso de Uso. Estos usualmente descritos en textos y corresponden de una representación textual del diagrama de secuencia.

2.2.7.2. Simbología para realizar los casos de uso

La simbología nos permite tener una fácil comprensión tanto para el diseñador del software como para los usuarios, a continuación presentamos la simbología para realizar los casos de uso.

Elementos.- Los elementos que pueden aparecer en un diagrama de casos de uso son: actores, casos de uso y relaciones entre casos de uso.

Actores.- Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.).



Figura 2.4. Usuario

Fuente: (Recuperado 01 de junio del 2013) PowerDesigner 12.

Tipos de actores.

- ✓ Principales.- Personas que mantienen o administran el sistema.
- ✓ Secundarios.- Personas que usan el sistema.
- ✓ Material externo.- Dispositivos materiales imprescindibles que formen parte del ámbito de la aplicación.
- ✓ Otros sistemas.- Sistemas con que el sistema interactúa.

Casos de uso.- Es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el diagrama de casos de uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

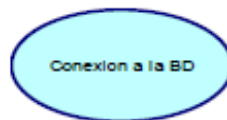


Figura 2.5. Caso de uso

Fuente: (Recuperado 01 de junio del 2013) PowerDesigner 12.

Relaciones entre casos de uso.- Entre dos casos de uso puede haber las siguientes relaciones:

Extiende: Cuando un caso de uso especializa a otro extendiendo su funcionalidad.

Usa: Cuando un caso de uso utiliza a otro.



Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<usa>> según sea el tipo de relación.

Asociación.- Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación. Dicha relación se denota con una flecha simple



Figura 2.6. Asociación

Fuente: (Recuperado 01 de junio del 2013) PowerDesigner 12.

Dependencia.- Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.



Figura 2.7. Dependencia

Fuente: (Recuperado 01 de junio del 2013) PowerDesigner 12.

Generalización.- Cumple una doble función dependiendo de su estereotipo, que puede ser de uso (<<usa>>) o de herencia (<<extiende>>) este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).



Figura 2.8. Generalización

Fuente: (Recuperado 01 de junio del 2013) PowerDesigner 12.

Diagramas de interacción.- En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

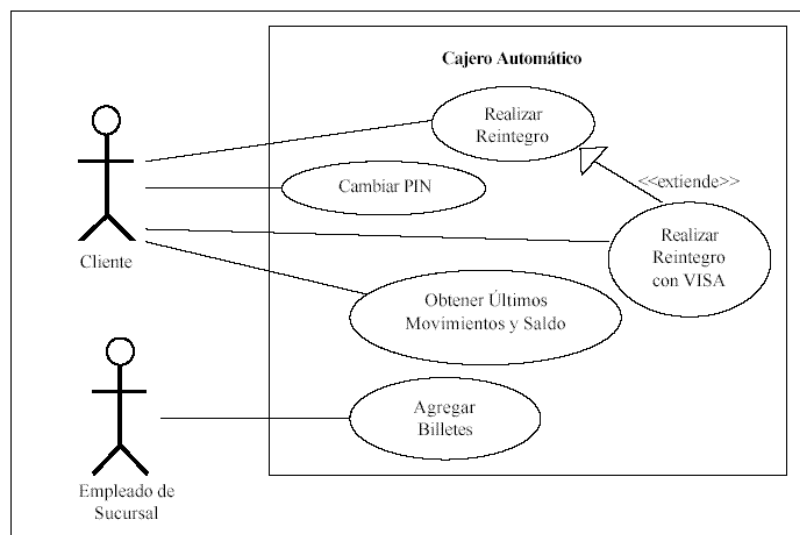


Figura 2.9. Ejemplo de diagrama de caso de uso

Fuente: (Recuperado 01 de junio del 2013) PowerDesigner 12.

2.2.7.3. Identificación de actores en la aplicación.

La primera aproximación es la identificación a los actores que interactúan con la aplicación. Se ha detectado dos tipos de usuario dentro del sistema de inscripción y matriculación. Estos se detallan a continuación:

- ✓ Administrador.- Es el usuario principal, persona que interactúa siempre con la totalidad de la aplicación, realizando operaciones como: Creación de usuarios, ingreso de información, generación de reportes, emisión de los diferentes certificados, visualizar reportes y es el encargado de administrar la información proveniente de cada una de las personas.
- ✓ Usuario.- Es un usuario secundario, realiza operaciones como: ingreso datos, emite certificados, genera reportes y visualizar reportes.

2.2.7.4. Diagrama de secuencia.

Un diagrama de secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor



tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo.

Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

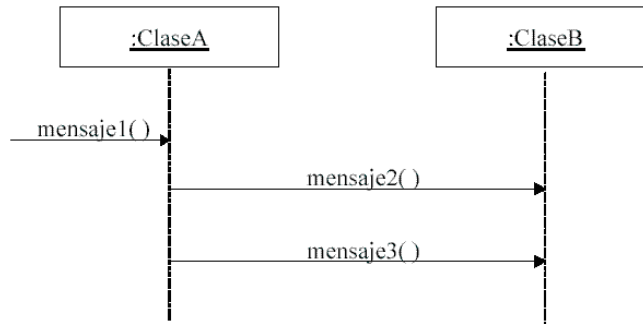


Figura 2.10. Ejemplo de diagrama de secuencia

Objeto (actor).- La representación gráfica es un rectángulo, que representa una instancia de un objeto en particular.

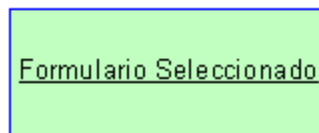


Figura 2.11. Objeto

Eje vertical (línea punteada).- Representa el tiempo que se coloca las llamadas a métodos del objeto sin un orden prefijado.

El tiempo fluye de arriba hacia abajo, se colocan etiquetas, descripciones de actividades.



Figura 2.12. Diagrama eje vertical

Mensaje al mismo objeto.- Es una llamada al objeto externo, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio.

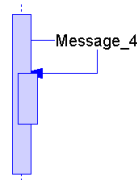


Figura 2.13. Mensaje al mismo objeto

Mensaje de un objeto a otro objeto.- Se representa con una flecha entre un objeto y otro, representa la llamada a un método (operación) de un objeto en particular.



Figura 2.14. Mensaje de un objeto a otro objeto

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior.

Cada objeto representa una columna distinta. Se pone un símbolo de objeto al final de la flecha que representa el mensaje que ha creado el objeto.

El orden relativo de los objetos no tiene significado aun cuando resulta útil organizarlo de modo que se minimice la distancia de las flechas.

2.2.8. Software libre

El software libre (free software) respeta la libertad de los usuarios sobre el producto que éste adquiera, por lo que, una vez obtenido, éste puede ser distribuido, modificado, redistribuido, copiado y usado libremente.

Este tipo de software se basa en cuatro libertades:

- ✓ Libertad para usarlo con cualquier propósito.
- ✓ Libertad para modificarlo a nuestras necesidades.
- ✓ Libertad para distribuir copias y
- ✓ Libertad para mejorarlo.



Éste software generalmente está disponible en forma gratuita o al precio de costo de distribución, sin embargo no es obligatorio que sea así, por lo tanto no se debe asociar software libre con "software gratuito", ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" incluye en ocasiones el código fuente; sin embargo, éste tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

2.2.8.1. Ventajas del software libre

- ✓ Bajo costo de adquisición: Se trata de un software económico ya que permite un ahorro de grandes cantidades en la adquisición de las licencias.
- ✓ Innovación tecnológica: esto se debe a que cada usuario puede aportar sus conocimientos y su experiencia y así decidir de manera conjunta hacia donde se debe dirigir la evolución y el desarrollo del software.
- ✓ Independencia del proveedor: al disponer del código fuente, se garantiza una independencia del proveedor que hace que cada empresa o particular pueda seguir contribuyendo al desarrollo y los servicios del software.
- ✓ Escrutinio público: esto hace que la corrección de errores y la mejora del producto se lleven a cabo de manera rápida y eficaz por cada uno de los usuarios que lleguen a utilizar el producto.
- ✓ Adaptación del software: esta cualidad resulta de gran utilidad para empresas e industrias específicas que necesitan un software personalizado para realizar un trabajo específico y con el software libre se puede realizar y con costes totales de operación (TCO) mucho más razonables.

2.2.9. Lenguaje de Programación Java

Java fue diseñado en 1990 por James Gosling, de Sun Microsystems, como software para dispositivos electrónicos de consumo. Curiosamente, todo este lenguaje fue



diseñado antes de que diese comienzo la era World Wide Web⁵, puesto que fue diseñado para dispositivos electrónicos como calculadoras, microondas y la televisión interactiva.

Inicialmente Java se llamó Oak (roble en inglés), aunque tuvo que cambiar de denominación, debido a que dicho nombre ya estaba registrado por otra empresa. Se dice este nombre se le puso debido a la existencia de tal árbol en los alrededores del lugar de trabajo de los promotores del lenguaje.

Tres de las principales razones que llevaron a crear Java son:

- ✓ Creciente necesidad de interfaces mucho más cómodas e intuitivas que los sistemas de ventanas que proliferaban hasta el momento.
- ✓ Fiabilidad del código y facilidad de desarrollo. Gosling observó que muchas de las características que ofrecían C o C++ aumentaban de forma alarmante el gran coste de pruebas y depuración. Por ello en los sus ratos libres creó un lenguaje de programación donde intentaba solucionar los fallos que encontraba en C++.
- ✓ Enorme diversidad de controladores electrónicos. Los dispositivos electrónicos se controlan mediante la utilización de microprocesadores de bajo precio y reducidas prestaciones, que varían cada poco tiempo y que utilizan diversos conjuntos de instrucciones. Java permite escribir un código común para todos los dispositivos.

El proyecto Green fue el primero en el que se aplicó Java, y consistía en un sistema de control completo de los aparatos electrónicos y el entorno de un hogar. Con este fin se construyó un ordenador experimental denominado *7 (StarSeven).

El sistema presentaba una interfaz basada en la representación de la casa de forma animada y el control se llevaba a cabo mediante una pantalla sensible al tacto. En el sistema aparecía ya Duke, la actual mascota de Java.

⁵(Recuperado el 01 de diciembre del 2012) de http://es.wikipedia.org/wiki/World_Wide_Web



Figura 2.15.Duke, la mascota de Java

Fuente: (Recuperado 01 de diciembre del 2012) de

http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29

2.2.9.1. Características del lenguaje Java

Lenguaje simple: Java posee una curva de aprendizaje muy rápida. Resulta relativamente sencillo escribir applets interesantes desde el principio. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros. Debido a su semejanza con C y C++, y dado que la mayoría de la gente los conoce aunque sea de forma elemental, resulta muy fácil aprender Java. Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo.

Orientado a objetos: Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

Distribuido: Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador.

Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).

Robusto: Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.



Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

Seguro: Dada la naturaleza distribuida de Java, donde las applets se bajan desde cualquier punto de la Red, la seguridad se impuso como una necesidad de vital importancia. A nadie le gustaría ejecutar en su ordenador programas con acceso total a su sistema, procedentes de fuentes desconocidas. Así que se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.

Indiferente a la arquitectura: Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución tan variopintos, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura diseñada para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de problemas los soluciona el intérprete de Java.

Portable: La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas. Estas dos últimas características se conocen como la Máquina Virtual Java (JVM).

Alto rendimiento

Multihebra: Hoy en día ya se ven como terriblemente limitadas las aplicaciones que sólo pueden ejecutar una acción a la vez. Java soporta sincronización de múltiples hilos de ejecución a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.



Dinámico: El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red.

Produce applets: Java puede ser usado para crear dos tipos de programas: aplicaciones independientes y applets. Las aplicaciones independientes se comportan como cualquier otro programa escrito en cualquier lenguaje, como por ejemplo el navegador de Web HotJava. Por su parte, las applets son pequeños programas que aparecen embebidos en las páginas Web, como aparecen los gráficos o el texto, pero con la capacidad de ejecutar acciones muy complejas, como animar imágenes, establecer conexiones de red, presentar menús y cuadros de diálogo para luego emprender acciones.

2.2.10. IDE para Programación en Java: Eclipse

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) .

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project⁶, cubriendo casi todas las áreas de Model Driven Engineering⁷.

2.2.11. Aplicaciones web con JPA, EJB, JSF y PrimeFaces

Con el avance de las tecnologías web y la Internet, se han abierto nuevas oportunidades para los desarrolladores de aplicaciones empresariales; permitiéndoles

⁶(Recuperado el 01 de diciembre del 2012) de http://es.wikipedia.org/w/index.php?title=Eclipse_Modeling_Project&action=edit&redlink=1

⁷(Recuperado el 01 de diciembre del 2012) de http://es.wikipedia.org/wiki/Model_Driven_Engineering



el uso de las nuevas tecnologías web en el desarrollo de aplicaciones mucho más robustas, escalables y con un mayor rendimiento. Algunas de las nuevas tecnologías que han surgido son: JavaServer Faces que es la tecnología estándar de la edición empresarial de Java EE para la creación de interfaces de usuario en la web y que permite integrar otras tecnologías como las hojas de estilo en cascada (CSS) que describen como se va a mostrar un documento; un modelo de desarrollo web para crear aplicaciones interactivas, JavaBeans empresariales (EJB) y el API de Java para el manejo de entidades persistentes (JPA) sobre bases de datos relacionales.

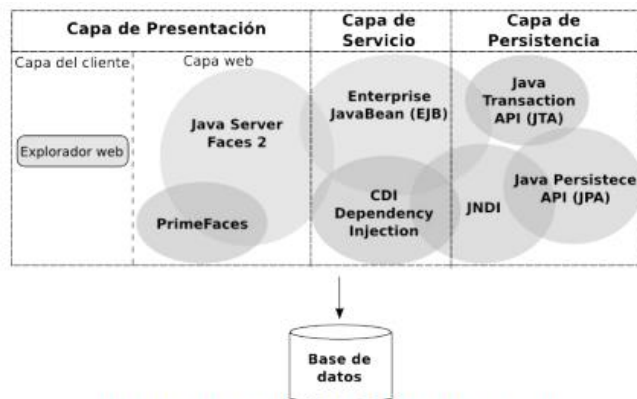


Fig. 1.1. Tecnologías Java EE para el desarrollo de aplicaciones web.

Figura 2.16.. Tecnologías Java EE para el desarrollo de aplicaciones web

Fuente: (Recuperado 01 de diciembre del 2012) de

<http://www.tamps.cinvestav.mx/~fpech/sd/files/paper001.pdf>

Java Enterprise Edition (Java EE)

Proporciona un conjunto de especificaciones técnicas para el desarrollo de aplicaciones empresariales, facilitar el desarrollo de aplicaciones distribuidas, robustas, potentes y de alta disponibilidad.

Java EE define cuatro tipos de componentes:

- ✓ Applets: Aplicaciones GUI que se ejecutan en un navegador.
- ✓ Aplicaciones: Son programas que se ejecutan en un cliente
- ✓ Aplicaciones web: (servlets, páginas JSP y JSF) Se ejecutan en un contenedor web y responden a las peticiones HTTP del cliente.
- ✓ Aplicaciones Empresa:(EJB, JMS, JTA, etc.) Son ejecutadas en un contenedor EJB.



Java EE se divide en dominios lógicos llamados contenedores. Cada contenedor tiene una función específica, soporta un conjunto de APIs y ofrece servicios a los componentes tales como seguridad, acceso a base de datos, gestión de transacciones, nombres de directorios, e inyección de recursos. Los contenedores ocultan la complejidad técnica y mejoran la portabilidad. El contenedor EJB es responsable de administrar la ejecución de los beans⁸ que contiene la lógica de negocio.

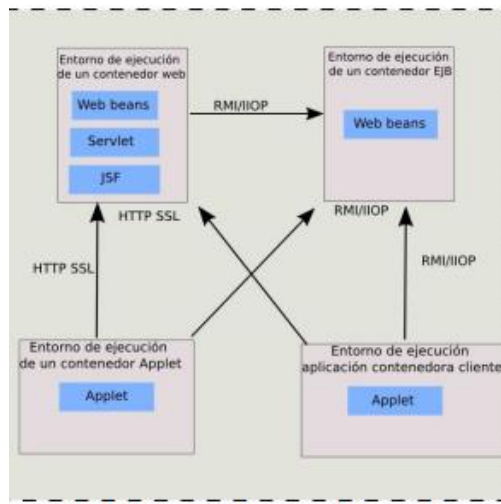


Figura 2.17. Estándar de los contenedores Java EE

Fuente: (Recuperado 01 de diciembre del 2012) de

<http://www.tamps.cinvestav.mx/~fpech/sd/files/paper001.pdf>

Los contenedores proporcionan servicios subyacentes a sus componentes desplegados, esto permite al desarrollador centrarse en la lógica de aplicación en lugar de resolver problemas técnicos.

Algunos de los servicios que proporciona Java EE se describen a continuación:

- ✓ Java Transaction API⁹ (JTA): Este servicio ofrece una demarcación de transacciones API utilizada por el contenedor y la aplicación. También proporciona una interfaz entre el administrador de transacciones y el administrador de recursos en el nivel Service Provider Interface (SPI).

⁸(Recuperado el 01 de diciembre del 2012) de <http://es.wikipedia.org/wiki/Bean>

⁹(Recuperado el 01 de diciembre del 2012) de http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones



- ✓ Java Persistence API (JPA): API estándar para el mapeo de objeto-relacional (ORM). Con Java Persistence Query Language (JPQL), se puede consultar objetos almacenados en la base de datos subyacente.
- ✓ Validación: El Bean de validación proporciona un nivel de declaración de restricción de la clase y la facilidad de validación.
- ✓ Java Message Service (JMS): Permite que los componentes se comuniquen de forma asincrónica a través de mensajes.
- ✓ Java Naming and Directory Interface (JNDI): Esta API, incluida en Java SE, se utiliza para acceder a los sistemas de nombres y directorios.
- ✓ JavaMail: Muchas aplicaciones requieren la capacidad de enviar correos electrónicos que pueden ser implementadas a través del uso de la APIJavaMail

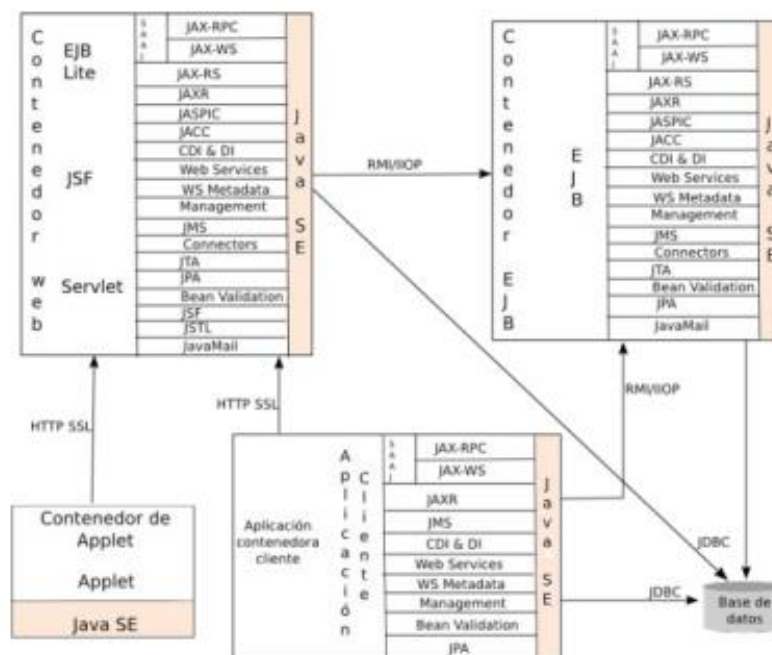


Figura 2.18. Servicios proporcionados por los contenedores

Fuente: (Recuperado 01 de diciembre del 2012) de

<http://www.tamps.cinvestav.mx/~fpech/sd/files/paper001.pdf>

Java Server Faces (JSF)

Es un estándar de Java hacia la construcción de interfaces de usuario para aplicaciones web que simplifican el desarrollo de aplicaciones web del lado del cliente, JSF está basado en la tecnología Java EE. En el 2009 se dio a conocer la nueva versión JSF 2.0, que contiene algunas características y/o mejoras con respecto



a las versiones anteriores (JSF 1.0, JSF 1.1 y JSF 1.2) como son: Mejoras en la navegación: navegación condicional, inspección en tiempo de ejecución en las reglas de navegación. Control de excepciones: permite fácilmente la creación de una página de error que utiliza componentes JSF. Mejoras en la expresión del lenguaje: compatibilidad con métodos arbitrarios incluyendo el paso de parámetros. Validación: es una nueva especificación java desarrollada para la validación de beans.

Una página JSF utiliza la extensión *.xhtml, es decir, una combinación de XML con HTML y puede incluir componentes como CSS, JavaScript, entre otros.

La especificación de JSF define seis fases distintas en su ciclo de vida:

- ✓ Restauración de la vista: Crea un árbol de componentes en el servidor para representar la información de un cliente.
- ✓ Aplicar valores de la petición: Actualiza los valores del servidor con datos del cliente.
- ✓ Proceso de validación: Valida los datos del usuario y hace la conversión.
- ✓ Actualización de valores del modelo: Actualiza el modelo del servidor con nuevos datos.
- ✓ Invocar la aplicación: Ejecutar cualquier lógica de aplicación para cumplir con la solicitud.
- ✓ Procesar la respuesta: Guarda un estado y da una respuesta al cliente.

Java Persistence API (JPA)

Fue creado con JEE para resolver problemas de persistencia de datos. Proporciona un modelo de persistencia para mapear bases de datos relacionales. En java EE 6, JPA 2.0 sigue el mismo camino de simplicidad y robustez y agrega nuevas funcionalidades. Se puede utilizar para acceder y manipular datos relacionales de Enterprise Java Beans (EJBs), componentes web y aplicaciones Java SE.

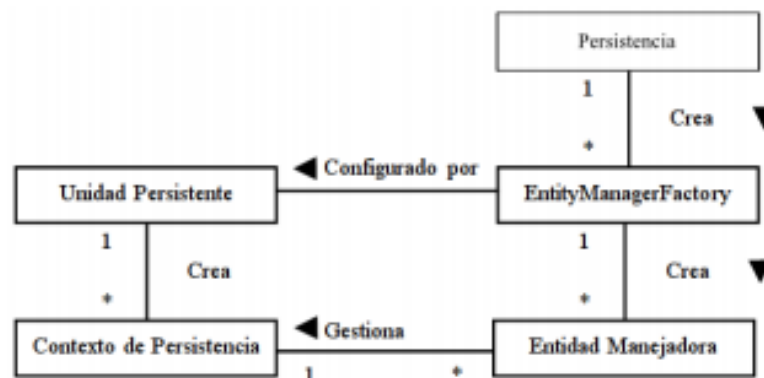


JPA es una abstracción que está por encima de JDBC lo que permite ser independiente de SQL. Todas las clases y anotaciones de esta API se encuentran en el paquete `javax.persistence`.

Los principales componentes de JPA son:

- ✓ Mapeo de base de datos relacionales (ORM). Es el mecanismo para mapear objetos a los datos almacenados en una base de datos relacional.
- ✓ Un API administrador de entidad para realizar operaciones en la base de datos tales como crear, leer, actualizar, eliminar (CRUD).
- ✓ El Java Persistence Query Language (JPQL) que permite recuperar datos con un lenguaje de consultas orientado a objetos.
- ✓ Las transacciones y mecanismos de bloqueo cuando se accedan a los datos concurrentemente, la API Java Transaction (JTA).

Una entidad es objeto de dominio de persistencia. Por lo general, una tabla en el modelo de datos relacional es representada por una entidad y sus instancias corresponden a los registros de dicha tabla. El estado de persistencia de una entidad es representado por propiedades persistentes, estas propiedades a su vez usan anotaciones para el mapeo de las entidades y relaciones entre entidades. Las relaciones entre entidades persistentes deben mapearse explícitamente como llaves foráneas o uniones de tablas, la manera en que se estructura una entidad, sus atributos y sus relaciones.





```
@Entity
@Table(name = "productos")
public class Productos implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "id_producto")
    private Integer claveProducto;
    @Basic(optional = false)
    @Column(name = "producto")
    private String descripcion;
    @Column(name = "class")
    private Integer claseProducto;
    @JoinColumn(name = "class", referencedColumnName = "clave_class")
    @ManyToOne(optional = false)
    private TipoProducto tipoProducto;
    @OneToMany(mappedBy = "productos")
    private Collection<Ventas> ventasCollection;
}
```

Definición de la entidad

Definición de los atributos

Definición de relaciones

Figura 2.19 .Definición de atributos y relaciones de una Entidad.

Fuente: (Recuperado 01 de diciembre del 2012) de

<http://www.tamps.cinvestav.mx/~fpech/sd/files/paper001.pdf>

EJB

Los JavaBeans empresariales son una tecnología que forma parte del estándar de Java EE. Están diseñados para desarrollo y despliegue de aplicaciones (distribuidas) de negocio basadas en componentes del lado del servidor. Una vez que se desarrolla una aplicación, ésta puede ser desplegada en cualquier servidor que soporte la especificación de EJB. Con esta tecnología es posible desarrollar aplicaciones empresariales sin tener que crear de nuevo los servicios de transacción, seguridad, persistencia, concurrencia y lo que se pueda necesitar en el proceso de creación de una aplicación; permitiendo a los desarrolladores enfocarse en la implementación de la lógica de negocio.

EJB divide la capa de negocio en dos partes: Capa de lógica de negocio donde se encuentra EJB y capa de persistencia. EJB cuenta con dos componentes de proceso de negocio, los beans de sesión (Session Beans) y los beans dirigidos por mensajes (Message-Driven Beans, MDBs), ambos son desarrollados por una aplicación de negocio e implementados y ejecutados por el Contenedor de EJB

PrimeFaces

Es una librería de componentes visuales de código abierto para el conjunto Java Server Faces 2.0 desarrollada y mantenida por Prime Technology. Su objetivo



principal es ofrecer un conjunto de componentes para facilitar la creación y diseño de aplicaciones web.

Los componentes de PrimeFaces cuentan con soporte nativo de Ajax, pero no se encuentra implícito, de tal manera que se tiene que especificar que componentes se deben actualizar al realizar una petición proporcionando así mayor control sobre los eventos. Cuenta también con un módulo adicional Touch Faces para el desarrollo de aplicaciones web para dispositivos móviles con navegadores basados en WebKit.

Las principales características de PrimeFaces son:

- ✓ Soporte nativo de Ajax, incluyendo Push/Coment.
- ✓ Kit para crear aplicaciones web móviles.
- ✓ Es compatible con otras librerías de componentes como Jboss RichFaces.
- ✓ Uso de JavaScript no intrusivo.
- ✓ Es un proyecto open source, activo y estable.

Estado actual de Eclipse

Eclipse se organiza en proyectos, estos proyectos a su vez dividen el trabajo en subproyectos, y los subproyectos en componentes, a continuación se enumeran los proyectos de alto nivel:

- ✓ Herramientas (ETP, Eclipse Tools Project). Herramientas varias y componentes comunes para la plataforma Eclipse.
- ✓ Web (WTP, Web Tools Project). Herramientas para el desarrollo de aplicaciones web y JEE (Java Enterprise Edition).
- ✓ Pruebas y rendimiento (TPTP, Test and Performance ToolsProject), herramientas de pruebas y medida de rendimientos para que los desarrolladores puedan monitorizar sus aplicaciones y hacerlas más productivas.
- ✓ Informes web (BIRT, Business Intelligence and Reporting Tools). Sistema de generación de informes web.
- ✓ Modelado (EMP, Eclipse Modeling Project). Herramientas de desarrollo basado en modelos.



- ✓ Datos (DTP, Data Tools Platform). Soporte para tecnologías centradas en el manejo de datos.
- ✓ Dispositivos empotrados (DSDP, Device Software Development Platform). Herramientas para el desarrollo de aplicaciones destinadas a ser ejecutadas en dispositivos limitados en hardware, esto es, dispositivos empotrados.
- ✓ Arquitectura Orientada a Servicios (SOA, Service Oriented Architecture). Herramientas para el desarrollo de proyectos orientados a servicios.
- ✓ Tecnología Eclipse. Investigación, divulgación y evolución de la plataforma Eclipse.

2.2.12. Servidor de Aplicaciones Jboss

El servidor JBOSS es un proyecto de código abierto, con el que se consigue un servidor de aplicaciones basado en J2EE, e implementado al 100% en Java.

El producto JBOSS es únicamente un "EJB Container" y es por esto que generalmente se utiliza en conjunción con un "Web-Container", en nuestro caso Tomcat.

Uno de los rasgos más importantes de JBOSS es su apoyo a la implementación "en caliente". Lo que esto significa es implementar un nuevo EJB es tan simple como copiar el archivo correspondiente en el directorio correspondiente. Si esto se hace mientras el Bean ya está cargado, JBOSS lo descarga automáticamente, y entonces carga la nueva versión.

Como se observa en el gráfico siguiente un "Java Application Server" se encuentra compuesto por dos partes: un "ServletEngine" y un "EJB Engine", dentro del "ServletEngine" se ejecutan exclusivamente las clásicas aplicaciones de un Servidor (JSP's y Servlets) , mientras el "EJB Engine(Container)" es reservado para aplicaciones desarrolladas alrededor de EJB's o Enterprise Java Bean

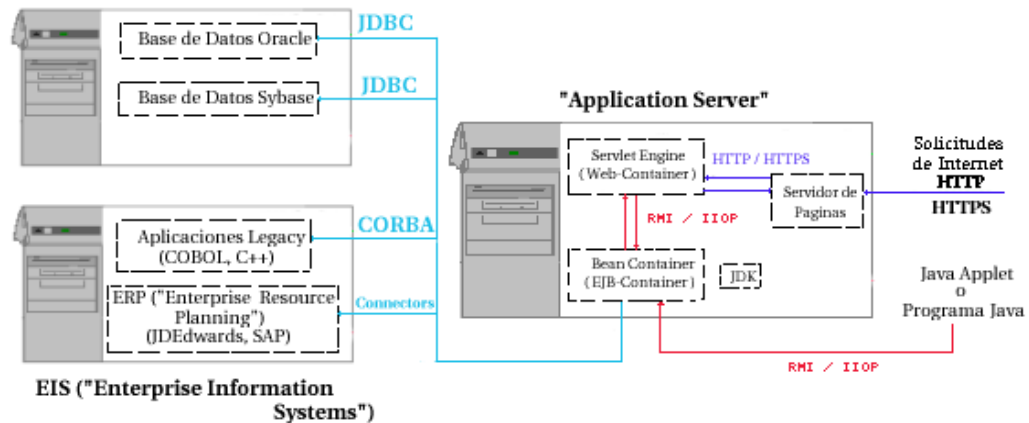


Figura 2.20. Triangulo del Proyecto

Fuente: (Recuperado el 07 de diciembre del 2012) de

<http://adolfoaraujo.wordpress.com/2012/09/14/apt07-triangulo-del-proyecto-y-su-calidad/>

JBOSS ha anunciado que su servidor de aplicaciones ha superado el Compatibility Test Suite y, por lo tanto, se ha convertido en el primer servidor de aplicaciones libre que obtiene la certificación de compatibilidad con J2EE 1.4.

La Estructura fundamental de JBOSS es la siguiente:

- ✓ **Bin:** Este directorio contiene los ejecutables utilizados por JBOSS, siendo el más importante el "script" de arranque utilizado por éste (run.sh).
- ✓ **Client:** Contiene los diversos archivos JAR's que serán utilizados por los distintos clientes de los EJB's utilizados en JBOSS. Dichos archivos deben ser agregados a la variable CLASSPATH del sistema donde radica el cliente.
- ✓ **Server:** Este directorio contiene tres sub-directorios nombrados: all, default y minimal; cada sub-directorio contiene los distintos archivos de configuración necesarios para ejecutar JBOSS en diferentes modalidades.

La ejecución de JBOSS es relativamente sencilla, dentro del directorio bin de la instalación de JBOSS se encuentran los archivos de arranque en forma de "scripts" para Shell. El archivo de ejecución run.sh inicia JBOSS con los parámetros encontrados en el directorio server/default/conf.



2.3. Identificación de las Variables

El concepto básico que todo administrador de proyectos debe de manejar es el referente al triángulo de administración de proyectos. Se trata de tener muy claro desde un principio cuál es el alcance del proyecto, el tiempo requerido y los recursos/presupuesto necesario para completarlo. Son los tres parámetros básicos con los que tendrá que lidiar el administrador de proyectos y que, al final, determinarán en gran medida si el proyecto fue o no exitoso.

Las variables se identificaran en base al estudio del triángulo de administración de proyectos. Los vértices de este triángulo son: tiempo, presupuesto y alcance. En el centro de este triángulo estaría la calidad del resultado final.



Figura 2.21. Triangulo del Proyecto

Fuente: (Recuperado el 07 de diciembre del 2012) de

<http://adolfoaraujo.wordpress.com/2012/09/14/apt07-triangulo-del-proyecto-y-su-calidad/>

El tiempo: El tiempo, es por lo tanto la delimitación más conocida. Todo proyecto está sujeto al tiempo, a una duración. La mayoría de los proyectos tienen una fecha límite para la que el proyecto deberá estar concluido. Además, el proyecto posiblemente disponga de una serie de hitos intermedios (o puntos intermedios de control) por cumplir en cuanto a fechas.

El coste: El coste, no solo es económico, puede ser humano (un electricista), de equipamiento (un ordenador o una pala excavadora), de material (ladrillos, cable, cinta de filmar), de instalaciones (alquiler de un estudio, de una sala de reuniones). Pero estos costes, al final y en definitiva se traducen en presupuesto económico. Para todos los proyectos, el coste supone una delimitación restrictiva. Solamente algunos proyectos no estarán sujetos a un presupuesto.



El ámbito: El ámbito del proyecto, también llamado alcance, es el trabajo requerido para realizarlo y conseguir el objetivo, el producto. Otra cosa bien distinta es el ámbito del producto (ámbito de proyecto frente a ámbito de producto) que es el conjunto de características, funciones, especificaciones y calidad final que tiene dicho producto una vez terminado. Los documentos (en forma de manuales) que esquematizan esta información sobre el producto, se denominan especificaciones del producto. El ámbito viene definido por las tareas a realizar.

Conocer la relación de la Calidad con el Triángulo del Proyecto

La calidad se encuentra en el centro del triángulo del proyecto. La calidad afecta a cada uno de los lados del triángulo y los cambios realizados en cualquiera de los lados del triángulo puede también afectar a la calidad. La calidad no es un factor propio del triángulo, sino el resultado de lo que se logra hacer con el tiempo, el dinero y el ámbito.

Por ejemplo, si descubre que dispone de más tiempo en la programación, puede aumentar el ámbito agregando tareas y duración. Con este tiempo y este ámbito adicionales, puede generar una mejor calidad en el proyecto y en las entregas.

O bien, si necesita recortar los costos para cumplir con el presupuesto, puede que necesite reducir el ámbito disminuyendo el número de tareas o la duración de las mismas. Con un ámbito reducido, puede que haya menos probabilidades de alcanzar cierto nivel de calidad. No obstante, una menor calidad es el resultado de la necesidad de reducir costos.

2.3.1. Variables independientes

Dentro de éste proyecto se ha determinado al alcance del proyecto como la variable independiente.

2.3.2. Variables dependientes

Una vez definida como independiente la variable “alcance del proyecto”, las variables “costo” y “tiempo” asociadas a dicho alcance corresponden a las variables dependientes.



2.4. Hipótesis

Una vez definido y dimensionado el alcance del proyecto; mismo que se diagramará con casos de uso, se definirá el costo y tiempo asociado al desarrollado de la tesis para lograr la integración de los módulos, planteada como plan piloto para el Patronato de Promoción Social de Rumiñahui.

Por tanto se quiere diseñar un sistema Web de inscripción y matriculación con el fin de facilitar al Patronato de Promoción Social de Rumiñahui las tareas manuales que se realiza al momento de la inscripción y la matriculación de los estudiantes a los cursos de capacitación.



CAPÍTULO 3

La finalidad de este capítulo es establecer el nivel de profundidad que se busca mediante el conocimiento propuesto, así como la forma de acceder a la información referente al estudio. Para cumplir con tal fin, el primer aspecto a definir es el diseño de la investigación, se escoge la población y muestra, se relacionan las técnicas e instrumentos, seguidamente se describe el procedimiento seguido para la realización del trabajo con el respectivo cronograma de actividades.

3. METODOLOGÍA

El desarrollo del proyecto de tesis comprende la definición de metodologías, a fin de aplicar mejores prácticas al estudio y desarrollo del proceso, y finalmente el planteamiento del proceso.

La metodología que va a ser utilizada en el desarrollo del Sistema Web para Inscripción a los cursos de Capacitación del Patronato de Promoción Social de Rumiñahui es la RUP/Easy (RationalUnifiedProcess/Easy), la misma que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo.

RUP/E es un conjunto de metodologías que se adaptan al contexto y a las necesidades de cada organización; que se centra en la arquitectura, siendo iterativa incremental y que es guiada por los casos de usos, basándose en el UML (UnifiedModelingLanguage) como herramienta principal.

3.1. Diseño de la Investigación

Para el desarrollo de esta investigación referida a un sistema de información se considerará para su diseño metodológico que de acuerdo a sus características, éste corresponde a un proyecto factible.

De acuerdo a los objetivos establecidos esta investigación basada en un Proyecto Factible, toma en cuenta el concepto dado por la Universidad Central de Ecuador, Director de Carrera de la Facultad de Ciencias Físicas y Matemática y Tutor, Guía para la Elaboración del Proyecto del Trabajo de Graduación, siendo éste:

El Proyecto Factible consiste en la investigación, elaboración y desarrollo de una un sistema informático adecuado a las necesidades del Patronato de Promoción



Social de Rumiñahui para el proceso de inscripción a los cursos de Capacitación, siendo el problema la falta de un sistema informático.

Consiste en recopilar datos valiéndose del manejo adecuado de libros, páginas Web, resultados de otras investigaciones, entrevistas, etc., por tal razón se usará para construir el marco científico y además para elaborar la propuesta de la presente tesis.

Pruebas de sistema. Una vez que se ha llegado a obtener una versión estable del sistema, se deben elaborar pruebas a éste para comprobar que todas las funcionalidades que componen el sistema están operando correctamente.

Pruebas de aceptación con la versión final del sistema. Una vez que el desarrollador prueban el correcto funcionamiento de la aplicación es turno de que el cliente apruebe el desarrollo de la aplicación, para esto se debe contar con la participación de las personas que administraran el sistema que evaluará la versión presentada con todas las funcionalidades y firmará un documento de aceptación del producto entregado.

Implantación en ambiente de producción. El sistema desarrollado deberá ser instalado en un servidor Web en una máquina del Patronato de Promoción Social.

Capacitación a usuarios expertos. Se brindarán capacitaciones a los usuarios encargados de realizar los diferentes procesos contemplados en éste sistema. Capacitación técnica. Existen usuarios dentro del Patronato de Promoción Social con formación técnica a quienes se debe indicar el funcionamiento del sistema entregado e implementado en la institución. Entrega formal del sistema. Una vez que el sistema ha sido implementado y sus usuarios capacitados en cuanto a sus roles, se debe firmar un documento en el que acepten que el producto entregado cumple con los requerimientos pactados y acepta su correcto funcionamiento.

3.1.1. Metodología RUP/ Easy

Durante los últimos años, una de las metodologías más populares ha sido el Rational



Unified Process (RUP). RUP, desarrollado por Rational Software Corporation, es un proceso de ingeniería de software que ofrece un enfoque disciplinado para asignar tareas y responsabilidades dentro de la organización del desarrollo. RUP captura algunas de las mejores prácticas de la industria para el desarrollo de software las cuales son para desarrollar el software en iteraciones, administrar requerimientos, usar arquitecturas basadas en componentes, verificar la calidad del software, controlar los cambios al software y modelar el software visualmente usando el Unified Modeling Language(UML). "El Unified Modeling Language (UML) es un método orientado a objetos y el lenguaje estándar de la industria para especificar, visualizar, construir y documentar los artefactos de sistemas de software."

3.1.2. Ventajas de la metodología RUP/Easy

Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.

- ✓ También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.
- ✓ La construcción de prototipos se puede utilizar como un modelo del proceso independiente, se emplea más comúnmente como una técnica susceptible de implementarse dentro del contexto de cualquiera de los modelos del proceso expuestos.
- ✓ Sin importar la forma en que éste se aplique, el paradigma de construcción de prototipos ayuda al desarrollador de software y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos.

3.2. Metodología de desarrollo

3.2.1. Modelo espiral

Es un modelo de proceso de software evolutivo que acompaña la naturaleza evolutiva de con los aspectos controlados y sistemáticos del ciclo de vida tradicional.



Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En este modelo, el sistema se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones se producen versiones cada vez más completas de ingeniería del sistema. .

El Modelo en Espiral se divide en un número de actividades estructurales, también llamadas "regiones de tareas". Generalmente existen entre tres y seis regiones de tareas:

- ✓ Comunicación con el cliente.- Las tareas requeridas para establecer comunicación entre el desarrollador y el cliente, sea revisar especificaciones, plantear necesidades.
- ✓ Planificación.- Las tareas requeridas para definir recursos, tiempos e información relacionada con el proyecto.
- ✓ Análisis de riesgos.- Las tareas requeridas para evaluar riesgos técnicos y de gestión.
- ✓ Ingeniería.- Las tareas requeridas para construir una o más representaciones de la aplicación
- ✓ Construcción y adaptación.- Las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- ✓ Evaluación del cliente.- Las tareas requeridas para obtener la reacción del cliente, según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación

El modelo en espiral es una de las metodologías¹⁰ más recomendables para el desarrollo y creación de un programa, ya que consta de pocas etapas o fases, las cuales se van realizando en una manera continua y cíclica.

Ciclo de vida - Modelo en espiral

El ciclo de vida de un proyecto es el periodo de tiempo transcurrido desde el inicio del análisis hasta el final de la etapa de transición de un proyecto

¹⁰(Recuperado el 01 de diciembre del 2013) de <http://definicion.de/metodologia/>



3.2.1.1. Fase de ingeniería

Esta fase involucra las etapas de análisis, diseño y planeación de un proyecto hasta el punto donde un plan de producción del sistema ha sido acordado y existe un entendimiento significativo del problema y de la solución. Se divide en las siguientes etapas:

- ✓ Fase de inicio o análisis.- en esta fase básicamente se determina la factibilidad del desarrollo, para ello se deben analizar los siguientes puntos:
- ✓ Técnica.- se analiza principalmente si existe la tecnología¹¹ para realizar el desarrollo del sistema que se pretende, además se debe verificar o evaluar la infraestructura informática necesaria para el desarrollo de la aplicación, es decir si se cuenta con el hardware, redes informáticas y accesos a internet.
- ✓ Economía (costo-beneficio).- se deben identificar y comparar los costos que trae consigo el proyecto y los beneficios asociados con éste proyecto. Para lograr éste fin es conveniente desglosar éste punto en las siguientes categorías:
- ✓ Costo de desarrollo.- Incluye los costos tangibles relacionados con la construcción del sistema tales como: salarios del equipo del proyecto (analistas, diseñadores, programadores), costo del software y hardware (lenguaje de programación, sistema operativo de red, componentes, licencias, servidores, impresoras, nuevo equipo de cómputo, etc.), honorarios de consultorios externos, capacitación. Los costos de desarrollo se realizan una sola vez durante el proyecto.
- ✓ Costos operacionales.- Son los costos que se requieren para operar el nuevo sistema, tales como salarios del personal operativo adicional, licencias de software, actualización de equipos, tarifas de comunicación, etc.

¹¹(Recuperado el 01 de diciembre del 2013) de <http://es.wikipedia.org/wiki/Tecnolog%C3%AD>



- ✓ Beneficios tangibles.- Son los ingresos adicionales y/o reducción de costos que el nuevo sistema proveerá, los beneficios tangibles pueden incluir aumento en ventas, reducción de pérdidas al mejorar el proceso, reducción de costos de inventario, etc.
- ✓ Beneficios intangibles.- Son beneficios extras que se pretenden obtener con el nuevo sistema, como el mejoramiento de servicio al cliente.

Una vez identificados los costos y beneficios se les deben de dar valores monetarios (Incluyendo los beneficios intangibles cuando sea posible).

- ✓ Organizacional.- Éste análisis estima que tan bien va a ser aceptado el sistema o los usuarios finales y que tan bien será incorporado a las operaciones actuales de la empresa.
- ✓ Requerimiento del sistema.- El primer paso en un nuevo proyecto de desarrollo es que alguien dentro de la empresa (gerente, agente de ventas, analista) visualice una oportunidad para mejorar el funcionamiento de la empresa. Los sistemas nuevos generalmente surgen de una necesidad o una oportunidad. En éste punto debemos identificar plenamente:
- ✓ Patrocinador del sistema.- Es la persona que tiene el interés de ver que el sistema funcione exitosamente, alguien que trabajará a lo largo del ciclo de vida del proyecto y que se asegurará de que siga el plan correcto desde el punto de vista de la empresa, debe también, tener la suficiente autoridad como para poder tomar decisiones relacionadas con el sistema. Esta persona será el contacto principal.
- ✓ Necesidad del negocio.- Describe porqué el sistema debe de ser desarrollado, las necesidades del negocio deben de ser claras y concisas pero en éste punto es probable que no estén completamente definidas.



- ✓ Funcionalidad del sistema.- Qué es lo que hará el nuevo sistema, delimitar las funciones que el sistema pretende cubrir.
- ✓ Valor monetario esperado.- Son las ganancias que se esperan obtener con el uso del nuevo sistema, estas se definen como tangibles e intangibles. El valor tangible puede ser cuantificado y medido (Ejemplo: deducción de costos operacionales). El valor intangible resulta de la creencia de que el sistema proveerá de beneficios importantes pero difíciles de cuantificar (Ejemplo: Mejor servicio al cliente).

3.2.1.2. Fase de elaboración

Al final de esta fase la “ingeniería” se considera completa. La decisión de comprometerse a producción es tomada. Las actividades de la fase de elaboración deben de asegurar que la arquitectura¹², requerimiento y los planes sean suficientemente estables y los riesgos suficientemente comprendidos, para que el costo y el calendario para la terminación del desarrollo pueda ser estimada dentro de un rango aceptable. La fase de elaboración se divide en las siguientes etapas:

- ✓ Entendimiento de los puntos críticos del sistema.-Tener un entendimiento más preciso de los puntos críticos que influyen en la arquitectura y la planeación del proyecto.
- ✓ Elaboración de la arquitectura y selección de componentes.- Establecer la arquitectura (estructura y funcionamiento básico del sistema). En esta etapa se toman las decisiones acerca de la compra de los componentes requeridos para la elaboración del sistema.
- ✓ Elaboración del prototipo.- Se elabora un prototipo rápidamente para demostrar que la arquitectura del sistema cumple los requerimientos de funcionalidad básica del sistema y que ataca los riesgos principales.

¹²(Recuperado el 01 de diciembre del 2012) de

<http://www.forosdelweb.com/f68/arquitectura-web-para-proyecto-php-911134/>



Aunque la meta siempre es la evolución del prototipo hasta un sistema funcional y confiable, no se debe excluir el desarrollo de uno o más prototipos exploratorios para mitigar riesgos específicos o para realizar demostraciones a los inversionistas.

3.2.1.3. Fase de producción

En la fase de producción se construyen versiones utilizables dentro de los planes anteriormente definidos que cumplen con los requerimientos del sistema establecidos en la fase de ingeniería. La fase de producción se divide en dos partes:

- ✓ Construcción.- Durante la fase de construcción todos los componentes y aplicaciones restantes son integradas al sistema, todas las funciones son minuciosamente probadas. El software recién creado es integrado cuando sea requerido. La fase de construcción representa un proceso de producción en donde se hace énfasis en administración de recursos y el control de operaciones para optimizar costos, tiempos y calidad. La fase de construcción incluye las siguientes etapas:
 - ✓ Administración, control y optimización de los recursos.
 - ✓ Desarrollo completo del sistema y prueba del cumplimiento de los requerimientos.
 - ✓ Elaboración de versiones beta.
- ✓ Transición.- La fase de transición inicia con el proyecto, alcanza una madurez suficiente para ser implementada en el dominio del usuario final. Éste normalmente requiere de una versión del sistema con calidad aceptable y documentación (manuales) para que la integración sea exitosa. Esta fase puede incluir cualquiera de las siguientes actividades:



- ✓ Prueba de versiones beta¹³ por parte de los usuarios finales en el ambiente real de trabajo.
- ✓ Prueba de corrida a paralelo, para realizar una corrida en paralelo es necesario realizar una conversión de datos previa (en caso de que el sistema actual sea automatizado) o una alimentación del sistema (en caso de que el sistema actual sea manual) de tal manera que ambos sistemas (el actual y el nuevo) inicien en el mismo punto. El objetivo de esta prueba es realizar las operaciones cotidianas de la empresa igualmente en ambos sistemas y comparar los resultados arrojados por ambos.

3.2.- Diseño Experimental

El diseño de investigación se cataloga por no experimental ya que no hay manipulación de variables, la acción de las variables se toman de la realidad y el investigador no interviene en ello, este diseño se observa desde el punto de vista transaccional, ya que permite el nivel de profundizar las variables en un único momento en el tiempo del tema investigado; así como a partir de datos que pueden ser obtenidos de otras fuentes disponibles.

La investigación se basa en la problemática que tiene el Patronato de Promoción Social de Rumiñahui en el proceso de inscripción y matriculación siendo uno de los problemas la falta de un sistema informático adecuado a sus necesidades.

Para la construcción de éste proyecto se plantea el seguimiento de un procedimiento que optimizará la construcción de éste sistema informático, ya que garantizara que se cumplirán todos los requerimientos impuestos por el cliente, involucrándolo directamente en cada etapa de desarrollo, para garantizar su conformidad con el producto final entregado. Además se desarrollara en herramientas de software libre.

El procedimiento se detalla a continuación:

¹³(Recuperado el 01 de diciembre del 2012) de <http://www.configurarequipos.com/doc806.html>



Realizar el levantamiento de requerimientos y expectativas del personal administrativo del Patronato de Promoción Social de Rumiñahui para ésta solución informática.

Definir qué procesos formarán parte de esta solución informática, así como los roles y responsable de su ejecución.

Mostrar cómo se realizan los procesos establecidos en un lenguaje universal que sea de fácil interpretación. Dichos procesos serán traducidos a diagramas UML para cumplir con éste fin.

Solicitar en el Patronato de Promoción Social de Rumiñahui la aprobación de los procesos, roles y el flujo de datos levantados para esta solución.

Construcción de la solución. Para éste punto, y por consecuencia directa del método en espiral escogido para el desarrollo de esta aplicación, se ejecutará la siguiente secuencia de pasos:

- ✓ Creación o actualización de un prototipo. Si es la primera iteración se debe crear la aplicación que será el primer referente del producto final y pasar al siguiente punto. De no ser así, se deben implementar los cambios recopilados durante la presentación del prototipo en la aplicación dentro de la versión actual. La funcionalidad agregada debe quedar lista para ser evaluada.
- ✓ Ejecución de pruebas unitarias de nuevas funciones. Aquí los desarrolladores prueban los cambios o nuevas funcionalidades añadidas a la aplicación.
- ✓ Presentación de prototipo. Se solicita al cliente evaluar la aplicación, para que éste informe si se deben realizar cambios a la misma.
- ✓ Recolección de observaciones y sugerencias realizadas por el Coordinador de los Cursos de Capacitación del Patronato de Promoción Social de Rumiñahui durante la presentación del prototipo y de ser necesario volver al primer punto, para empezar una nueva iteración.



Pruebas de sistema. Una vez que se ha llegado a obtener una versión estable del sistema, se deben elaborar pruebas a éste para comprobar que todas las funcionalidades que componen el sistema están operando correctamente.

Pruebas de aceptación con la versión final del sistema. Una vez que los desarrolladores prueban el correcto funcionamiento de la aplicación es turno de que el cliente apruebe el desarrollo de la aplicación, para esto se debe contar con la participación del Coordinador de los Cursos de Capacitación del Patronato de Promoción Social de Rumiñahui evaluará la versión presentada con todas las funcionalidades y firmará un documento de aceptación del producto entregado.

Implantación en ambiente de producción. El sistema desarrollado deberá ser instalado en el Patronato de Promoción Social de Rumiñahui y en un servidor Web.

Capacitación a usuarios expertos. Se brindarán capacitaciones a los usuarios encargados de realizar los diferentes procesos contemplados en éste sistema.

Capacitación técnica. Existen usuarios dentro del Patronato de Promoción Social de Rumiñahui con formación técnica a quienes se debe indicar el funcionamiento del sistema entregado e implementado en la institución.

Entrega formal del sistema. Una vez que el sistema ha sido implementado en el Patronato de Promoción Social de Rumiñahui y los usuarios capacitados en cuanto a sus roles, se debe firmar un documento el Coordinador de los Cursos de Capacitación del Patronato de Promoción Social de Rumiñahui y el Coordinador del Patronato de Promoción Social acepte que el producto entregado cumple con los requerimientos pactados y acepta su correcto funcionamiento.



3.3. Arquitectura MVC

El patrón de arquitectura MVC (Modelo Vista Controlador) es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del workflow de la aplicación). De esta forma, dividimos el sistema en tres capas donde, como explicaremos más adelante, tenemos la encapsulación de los datos, la interfaz o vista por otro y por último la lógica interna o controlador.

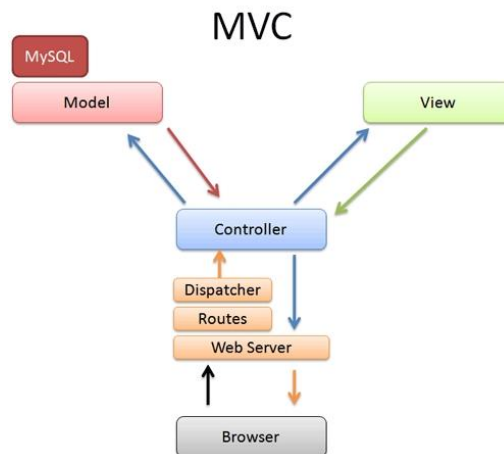


Figura 3.1. Arquitectura MVC.

Fuente: (Recuperado el 01 de diciembre del 2013) de

<http://www.rincondeloajeno.com/php-y-la-arquitectura-mvc/>

El patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones, compuesta por:

- ✓ Modelo

Contiene el núcleo de la funcionalidad (dominio) de la aplicación.

Encapsula el estado de la aplicación.

No sabe nada / independiente del Controlador y la Vista.

- ✓ Vista

Es la presentación del Modelo.

Puede acceder al Modelo pero nunca cambiar su estado.

Puede ser notificada cuando hay un cambio de estado en el Modelo.



✓ Controlador

Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente.

La unión entre capa de presentación y capa de negocio conocido en el paradigma de la programación por capas representaría la integración entre vista y su correspondiente controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la vista y el controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí. De hecho, este patrón separa el código en tres capas:

3.3.1. Capa modelo

Esta capa se encarga de interactuar con la base de datos y también se ejecuta las reglas de negocio.

3.3.2. Capa controlador

El controlador procesa las peticiones de la página web (vista), y envía estos datos a la capa modelo, para que esta le devuelva la información adecuada para mostrarla en la capa vista.

3.3.3. Capa vista

La vista es el código HTML que se muestra al usuario, con la información proveniente del controlador.

3.3.4. Tecnología WEB

Para el desarrollo de aplicaciones de negocio se utiliza frecuentemente el patrón de diseño MVC Modelo Vista Controlador que además es sencillo de implementar en las aplicaciones web. En este patrón el modelo es modificable por las funciones de negocio. Estas funciones son solicitadas por el usuario mediante el uso de un conjunto de vistas de la aplicación que solicitan dichas



funciones de negocio a través de un controlador, que es el módulo que recibe las peticiones de las vistas y las procesa. Se suele clasificar en dos tipos a las aplicaciones basadas en MVC:

- ✓ Tipo 1. Las vistas conocen la acción que se va a invocar en su petición, normalmente la función esta cableada dentro de la vista
- ✓ Tipo 2. El controlador introduce un conjunto de reglas que mapean a las peticiones con las funciones, controlando además el flujo de navegación por la aplicación.

3.3.5. ¿Por qué utilizar MVC?

El fácil mantenimiento de código en un futuro, ya que al estar separadas los distintos procesos según su tipo.

Si quisiéramos por ejemplo cambiar de tipo de base de datos, solo tendremos que cambiar la capa modelo.

3.3.6. Ventajas de MVC

- ✓ La separación del modelo de la vista es decir, separar los datos de la representación visual de los mismos.
- ✓ Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- ✓ Facilita agregar menos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- ✓ Crea independencia de funcionamiento.
- ✓ Facilita el mantenimiento en caso de errores.
- ✓ Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.
- ✓ Permite el escalamiento de la aplicación en caso de ser requerido.



CAPÍTULO 4

4. DIAGRAMACIÓN DEL SISTEMA

Diagramas UML

En base en las necesidades funcionales descritas previamente y levantadas en el capítulo anterior, se procede a exponer dos tipos de diagramas UML:

- ✓ Casos de uso
- ✓ Diagramas de secuencia.

4.1. Desarrollo del modelado del negocio

El Patronato de Promoción Social de Rumiñahui cuenta con un solo departamento, el cual maneja toda la información de los alumnos y cursos. El sistema representa el siguiente diagrama:

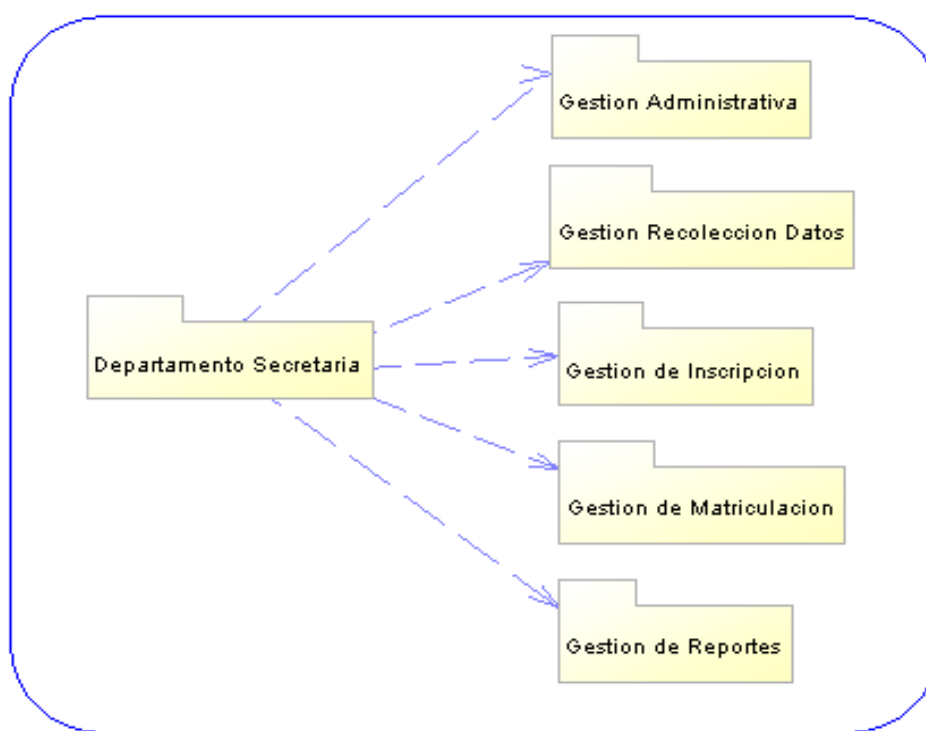


Figura 4.1. Modelado del negocio

Diseño de módulos

El sistema se compone de los siguientes módulos que son mencionados a continuación:



✓ Módulo de usuarios

En este módulo se realizará el control de acceso a la aplicación por medio de la autenticación de usuarios, que servirá para mantener seguro al sistema de accesos no autorizados.

✓ Módulo administrativo

El módulo administrativo constará de ABM de usuarios (Alta, Baja, Modificación de usuarios), así como los mantenimientos de información del sistema. Además en este módulo se encuentran opciones como:

- ✓ Opción de INSCRIPCIÓN es y matriculación: Deberá constar el ingreso de nuevos alumnos así como la modificación y desactivación de los mismos. También se deberá ingresar los cursos a los cual se matriculara.
- ✓ Opción eventos: Para la creación del evento debe tener información del lugar en donde se dictara el curso, instructor, cursos, capacidad y horarios
- ✓ Opción de reportes: Este módulo será el generador de reportes los cuales serán solicitados por los usuarios del sistema, según sus necesidades.

✓ Módulo Público

En este módulo el alumno se registra y puede realizar la inscripción para una vez que cancele el costo del curso pase de estado inscrito a matriculado.

4.2. Diagramas de casos de uso

El diagrama de casos de uso representa la forma como un cliente (actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).



ACTORES	ROL	ACCIONES PRINCIPALES
Administrador	Persona que solicita el inicio de trámite para ingresar al sistema.	Crear nuevos registros. Editar registros. Eliminar registros. Relacionar información. Realizar búsquedas.
Usuario	Persona que tiene acceso a limitados módulos.	Ingresa datos, visualizar información de cursos de capacitación.

Tabla 4.1. Actores y acciones principales

INVENTARIO DE CASOS DE USO

El sistema debe gestionar el nivel de acceso de los usuarios por medio de perfiles ya sea funcionario o administrador.

CODIGO	CASO DE USO	ACTORES
CU-001	Ingreso al sistema	Administrador, usuario
CU-002	Crear usuario	Administrador, sistema
CU-003	Modificar alumno	Administrador, sistema
CU-004	Inscripción alumno	Administrador, usuario, sistema
CU-005	Matricula alumno	Administrador, sistema
CU-006	Generar de reportes	Administrador, sistema

Tabla 4.2. Inventario de casos de uso.

4.2.1. Caso de uso: Ingreso al sistema.

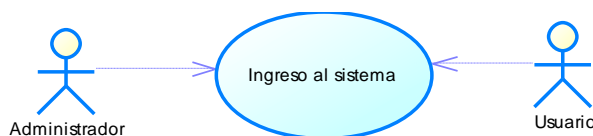


Figura 4.2. Ingreso al sistema.

Autor	Administrador del sistema, usuario
Descripción:	El usuario o administrador, intenta acceder al sistema.
Precondiciones:	El usuario debe encontrarse registrado dentro del sistema.
Pos-condiciones exitosas:	El usuario o administrador ingresa a una pantalla con un menú de opciones de acuerdo al rol con el que fue creado y podrá realizar los procesos correspondientes a su rol.
Pos-condiciones por falla:	Si el usuario no se encuentra registrado debe enviar una alerta que debe registrarse previamente.
Flujo normal:	<ol style="list-style-type: none"> 1. Validación del nombre de usuario y contraseña. 2. El sistema valida la existencia de los valores ingresados por el usuario. 3. Si los datos son correctos ingresa al perfil asignado para ese usuario. Caso contrario muestra un mensaje de error de autenticación. 4. Finaliza.



Flujo alternativo:	Existencia del usuario Si el usuario no existe lo notifica.
Excepciones:	Datos incorrectos. Muestra un mensaje de error de autenticación y no podrá ingresar al sistema.
Frecuencia de uso:	Cada vez que quiera ingresar al sistema.

Tabla 4.3. Caso de uso CU-001

4.2.2. Caso de uso: Administrador-Usuario-Creación de usuarios.

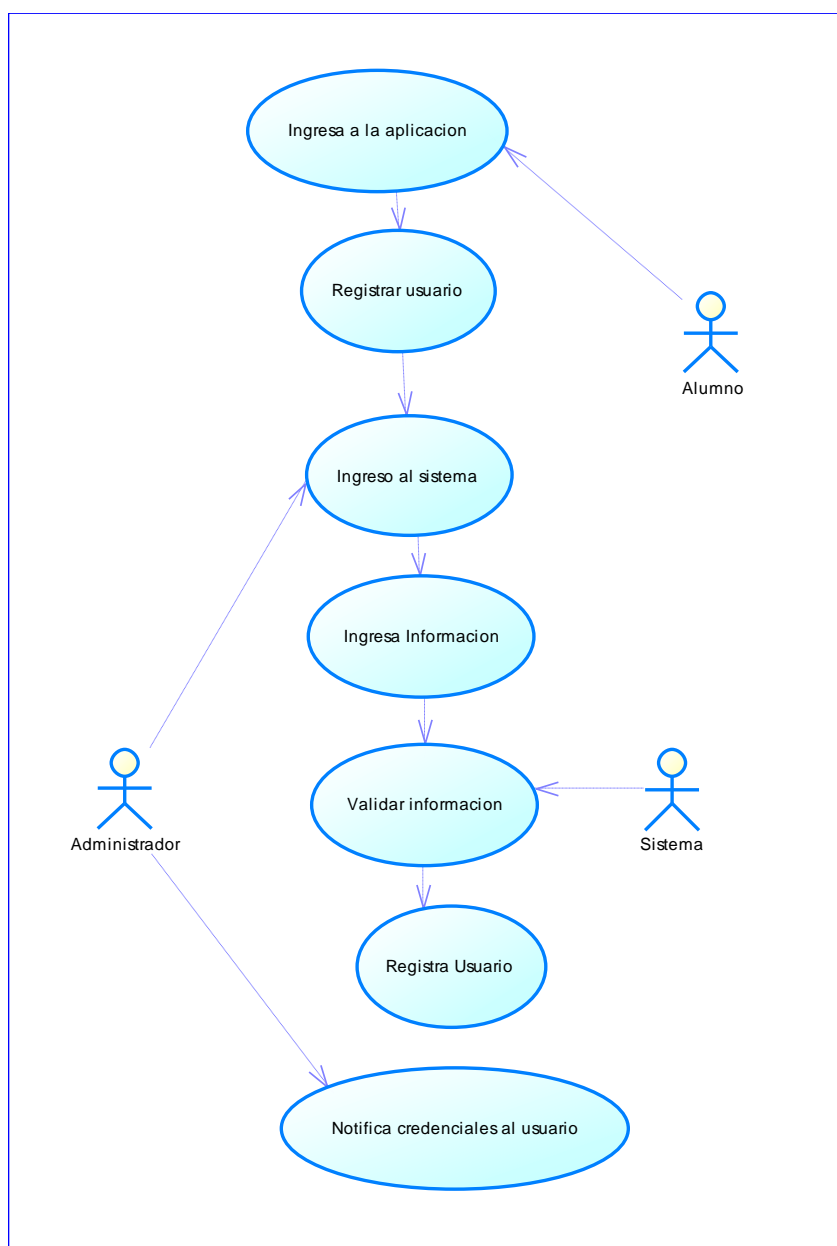


Figura 4.3. Creación de usuarios



Autor	Administrador del sistema, usuario
Descripción:	El alumno o administrador, crea un usuario
Precondiciones:	El administrador debe tener un usuario para ingresar al sistema y acceder a la opción de Alumno y crear un usuario. El alumno debe registrarse.
Pos-condiciones exitosas:	El administrador crea el usuario y da las credenciales al alumno. El alumno al registrarse crea su usuario.
Pos-condiciones por falla:	Se valida la información, si los datos ingresados son incorrectos o requeridos, presenta mensaje de error.
Flujo normal:	<ol style="list-style-type: none"> 1. Administrador ingresa al sistema. 2. Ingresa la información del alumno. 3. Sistema, valida la información 4. Registra Usuario. 5. Administrador notifica las credenciales usuario y contraseña al alumno.
Flujo alternativo:	<ol style="list-style-type: none"> 1. Alumno ingresa a la aplicación. 2. Alumno selecciona registrarse. 3. Ingresa información. 4. Sistema, valida la información. 5. Guarda 6. Alumno registrado, obtiene credenciales.
Excepciones:	Datos incorrectos. Muestra un mensaje de error de datos requeridos o datos incorrectos.
Frecuencia de uso:	Cada vez que quiera ingresar al sistema.

Tabla 4.4. Caso de uso CU-003

4.2.3. Caso de uso: Modificar alumno

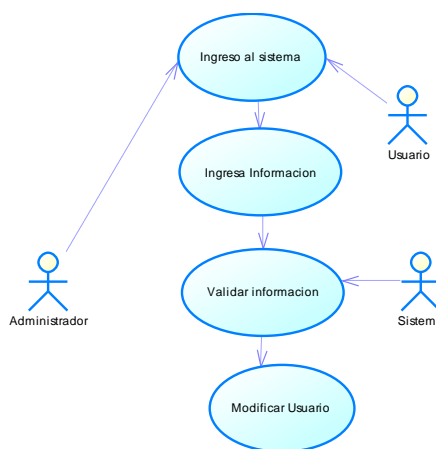


Figura 4.4. Modificar alumno



Autor	Administrador del sistema, usuario
Descripción:	El alumno o administrador, modificar datos
Precondiciones:	El administrador debe tener un usuario para ingresar al sistema y acceder a la opción de Alumno y editar datos. El alumno ingresa al sistema usando sus credenciales de usuario y contraseña para editar su información.
Pos-condiciones exitosas:	El administrador actualiza datos del alumno. El alumno edita su información
Pos-condiciones por falla:	Se valida la información, si los datos ingresados son incorrectos o requeridos, presenta mensaje de error.
Flujo normal:	<ol style="list-style-type: none"> 1. Administrador ingresa al sistema. 2. Buscar en el sistema al alumno. 3. Edita la información del alumno. 4. Sistema, valida la información 5. Edita alumno. 6. Administrador notifica las credenciales del alumno.
Flujo alternativo:	<ol style="list-style-type: none"> 1. Alumno ingresa al sistema con sus credenciales. 2. Edita información. 7. Sistema, valida la información. 8. Guarda
Excepciones:	Datos incorrectos. Muestra un mensaje de error de datos requeridos o datos incorrectos.
Frecuencia de uso:	Cada vez que quiera ingresar al sistema.

Tabla 4.5. Caso de uso CU-004



4.2.4. Caso de uso: Inscripción alumno

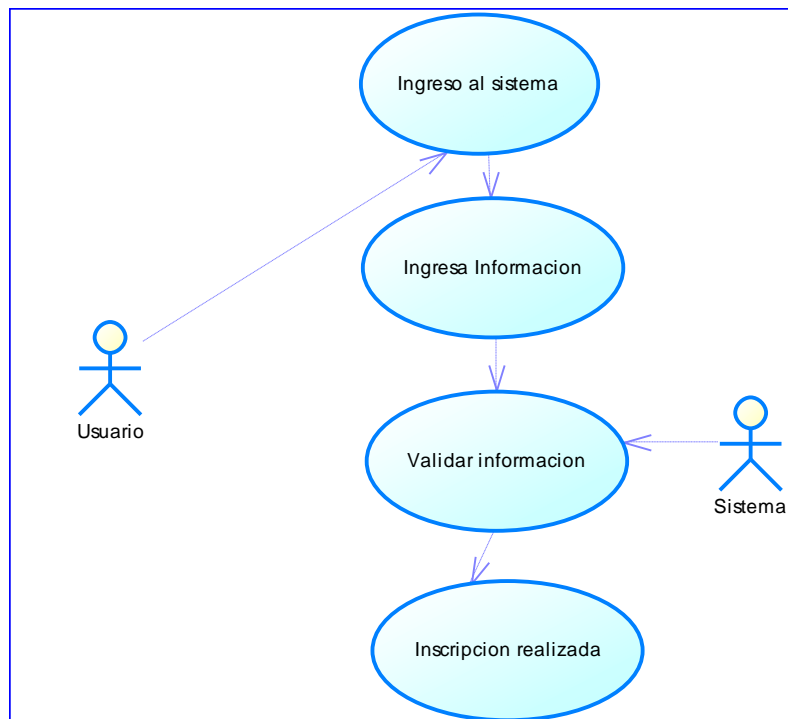


Figura 4.5. Inscripción alumno

Autor	usuario
Descripción:	El alumno se inscribe a un curso
Precondiciones:	El alumno debe estar registrado en el sistema.
Pos-condiciones exitosas:	El alumno selecciona el curso que desea inscribirse
Pos-condiciones por falla:	Se valida la información, si los datos ingresados son incorrectos o requeridos, presenta mensaje de error.
Flujo normal:	<ol style="list-style-type: none"> 1. Alumno ingresa al sistema. 2. Selecciona curso a inscribirse 3. Guarda información 4. Alumno inscrito
Flujo alternativo:	Todos los campos deben estar llenos
Excepciones:	Datos incorrectos. Muestra un mensaje de error de datos requeridos o datos incorrectos.
Frecuencia de uso:	Cada vez que quiera ingresar al sistema.

Tabla 4.6. Caso de uso CU-005



4.2.5. Caso de uso: Matriculación alumno

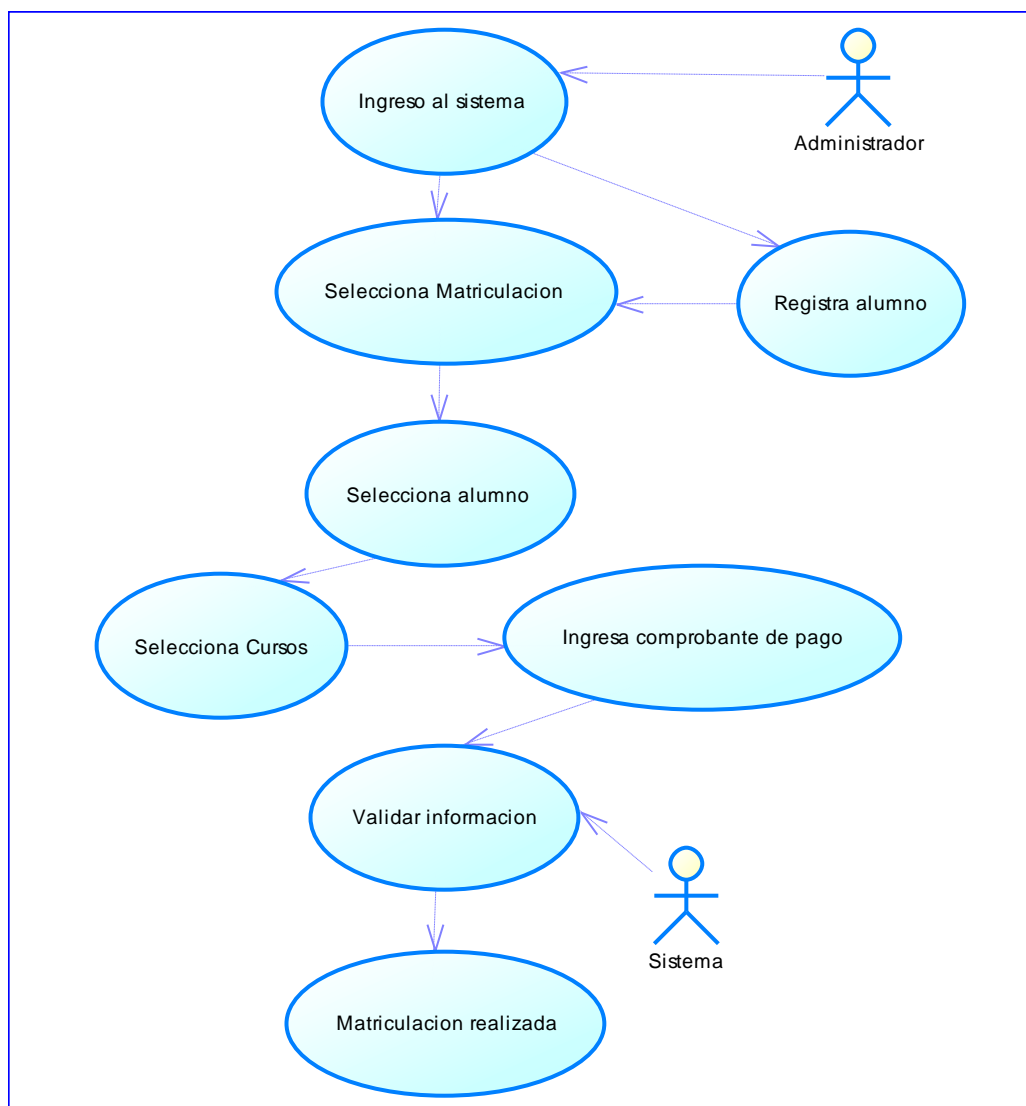


Figura 4.6. Matriculación alumno

Autor	Administrador, usuario
Descripción:	El administrador matricula un alumno
Precondiciones:	El alumno debe estar registrado en el sistema sino se encuentra debe el administrador registrarlo.
Pos-condiciones exitosas:	El administrador accede a las opciones para realizar el proceso matriculación.
Pos-condiciones por falla:	Se valida la información ingresada, si el alumno se inscribe en un mismo curso presenta mensaje que el alumno ya está inscrito en ese curso..
Flujo normal:	<ol style="list-style-type: none"> 1. Administrador ingresa al sistema. 2. Selecciona matriculación. 3. Selecciona alumno.



	4. Selecciona curso. 5. Confirma inscripción 6. Ingresa comprobante de pago 7. Alumno matriculado.
Flujo alternativo:	Todos los campos obligatorios deben estar llenos. Se recupera la información de la base de datos
Excepciones:	Datos incorrectos. Muestra un mensaje de error de datos requeridos o datos incorrectos.
Frecuencia de uso:	Cada vez que quiera ingresar al sistema.

Tabla 4.7. Caso de uso CU-006



4.3. Diagramas de Secuencia

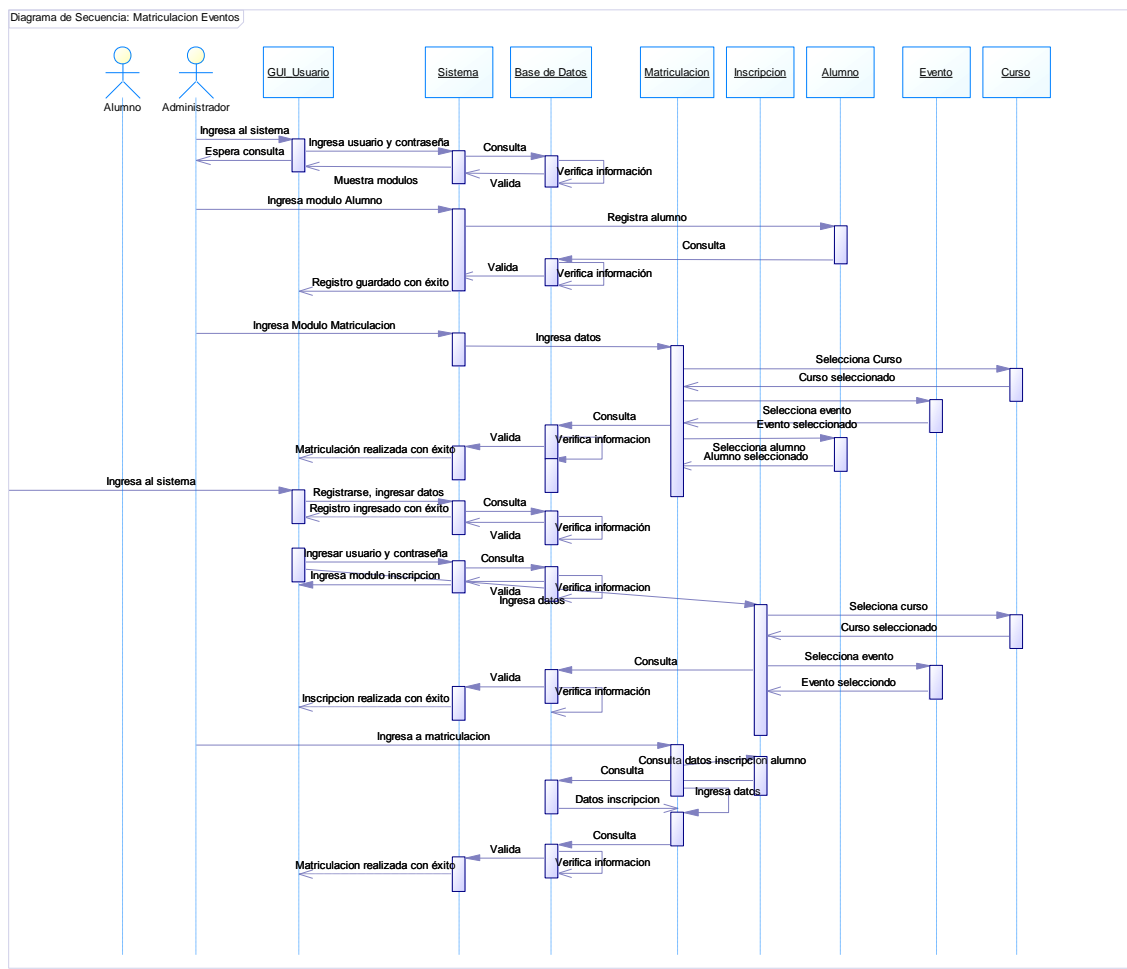


Figura 4.7. Diagrama de secuencia Sistema de Inscripción



CAPÍTULO 5

En este capítulo se detalla la instalación de las herramientas de desarrollo del “SISTEMA WEB PARA INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN DEL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUP” para su buen funcionamiento.

5. INSTALACIÓN DE HERRAMIENTAS

Para el desarrollo del presente proyecto de tesis se necesitan los siguientes requerimientos:

Sistemas operativos

- ✓ Clientes: Windows XP,7
- ✓ Servidor: Centos 6.5

Lenguaje de programación

- ✓ Html
- ✓ CSS
- ✓ JavaScript
- ✓ JQuery
- ✓ Ajax.

Base de datos

- ✓ Gestor de base de datos (MySQL).

Servidor de aplicaciones

- ✓ Jboss-6.1.0.Final

Ambiente de desarrollo

- ✓ Netbeans 7.1



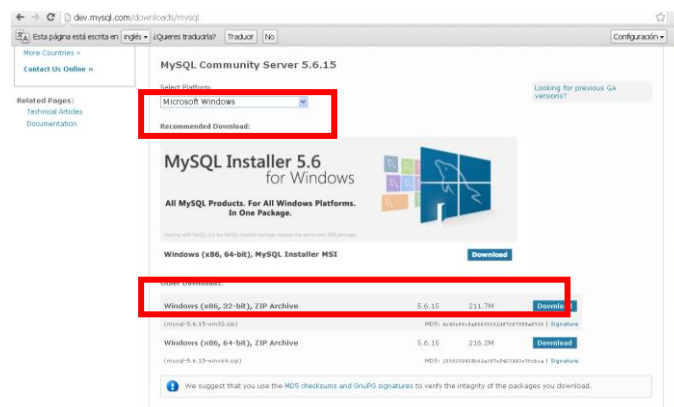
A continuación se presenta en forma detallada la instalación de cada una de las herramientas empleadas para el desarrollo del presente proyecto de tesis.

5.1. Herramientas para el desarrollo.

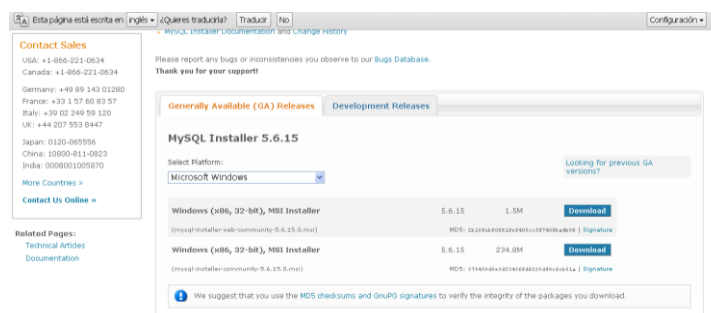
5.1.1 Instalar MySQL Community Server 5.6.12

En primer lugar descargaremos el fichero de instalación de MySQL Community Server 5.6.12, para ello accederemos a la web oficial de MySQL, en la URL: <http://dev.mysql.com/downloads/mysql>

Seleccionaremos el sistema operativo en "Select Platform", en nuestro caso "Microsoft Windows" y pulsaremos "Download" en "Windows (x86, 64-bit), MySQL Installer MSI":



Pulsaremos a continuación en "Download" en "Windows (x86, 32-bit), MSI Installer" (de 186MB):



Guardaremos el fichero que se descargará (mysql-installer-community-5.6.12.2.msi):



Ejecutaremos el fichero descargado.

Pulsaremos "Ejecutar" en la ventana de advertencia de seguridad:

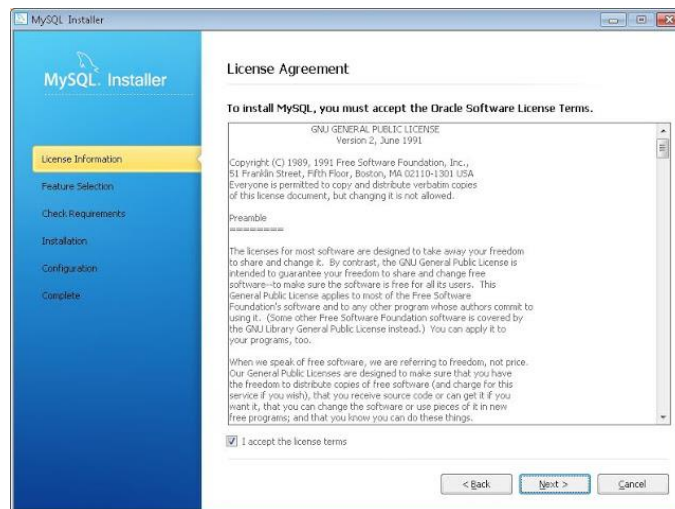


Se iniciará el asistente para instalar MySQL Community Server 5.6, pulsaremos en "Add/Modofy Products and Features":



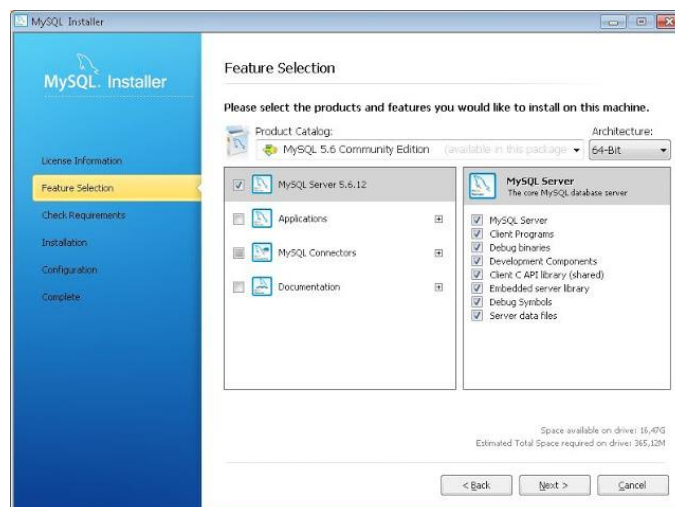


Leeremos los términos de licencia de MySQL Community Server, si estamos de acuerdo marcaremos "I accept the license terms" y pulsaremos "Next":



A continuación marcaremos las características a instalar para **MySQL Server 5.2.12**:

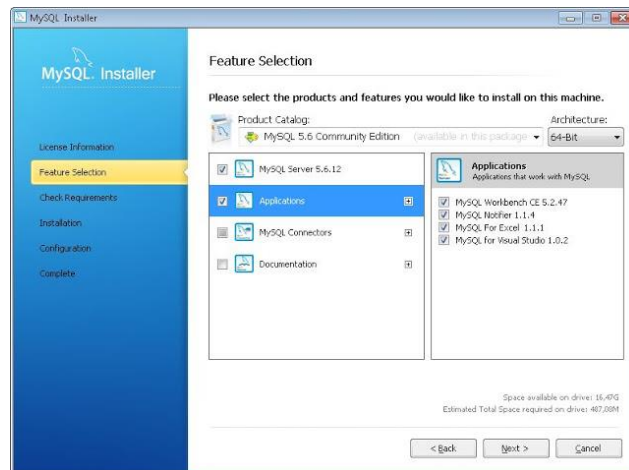
- ✓ MySQL Server.
- ✓ Client Programs.
- ✓ Debug binaries.
- ✓ Development Components.
- ✓ Cliente C API library (shared).
- ✓ Embedded server library.
- ✓ Debug Symbols.
- ✓ Server data files.





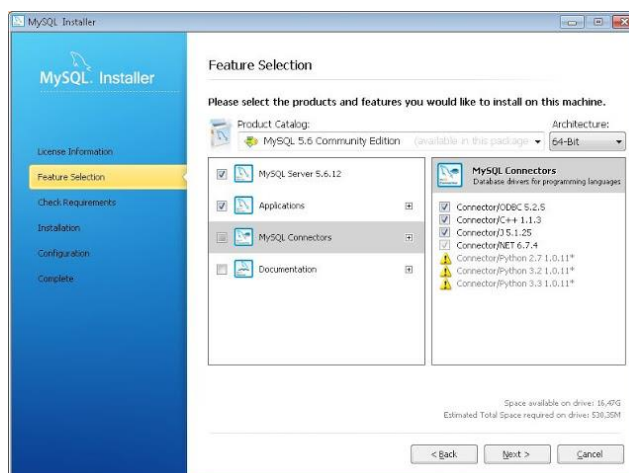
Para "Applications":

- ✓ MySQL Workbench CE 5.2.47.
- ✓ MySQL Notifier 1.1.4.
- ✓ MySQL for Excel 1.1.1.
- ✓ MySQL for Visual Studio 1.0.2.



Para "MySQL Connectors":

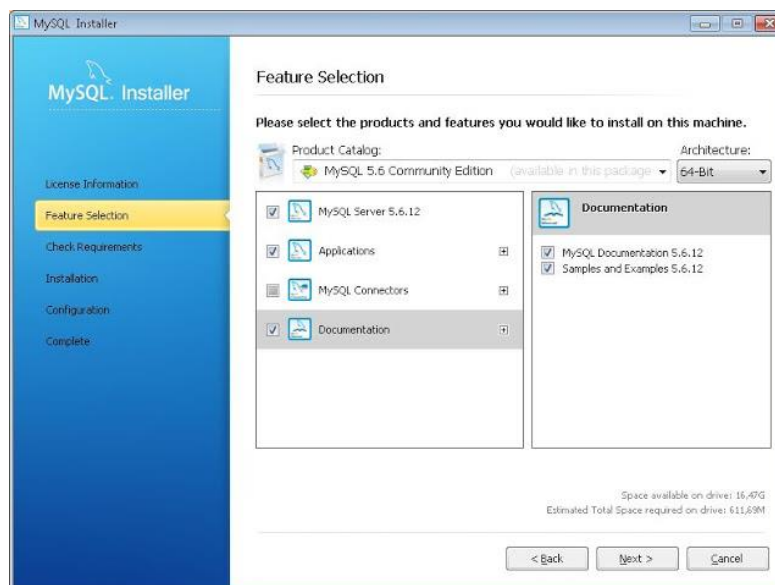
- ✓ Connector/ODBC 5.2.5.
- ✓ Connector/C++ 1.1.3.
- ✓ Connector/J 5.1.25.
- ✓ Connector/Net 6.7.4.
- ✓ Connector/Phyton 2.7.0.11.
- ✓ Connector/Phyton 3.2.1.0.11.
- ✓ Connector/Phyton 3.3.1.0.11.



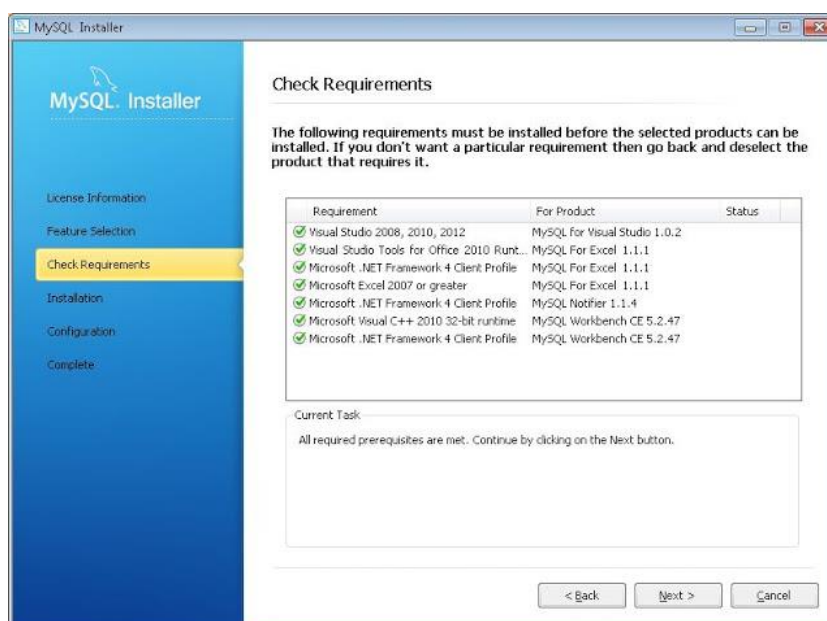


Para "Documentation":

- ✓ MySQL Documentation 5.6.12.
- ✓ Samples and Examples 5.6.12.

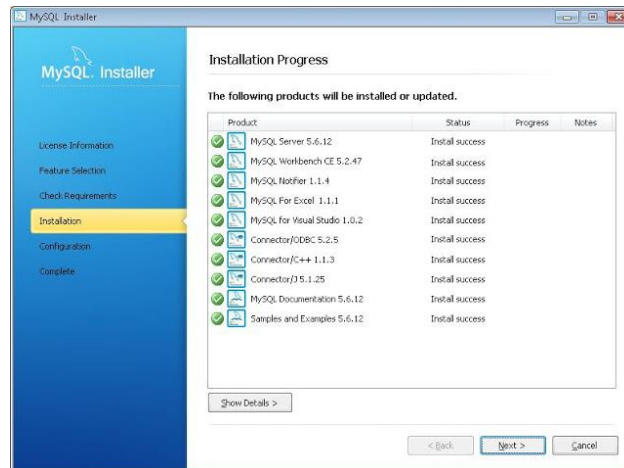


Según las características que hayamos elegido para instalar el asistente realizarán una comprobación de requisitos, a continuación mostrará una tabla de los requisitos para cada producto marcado para instalar:

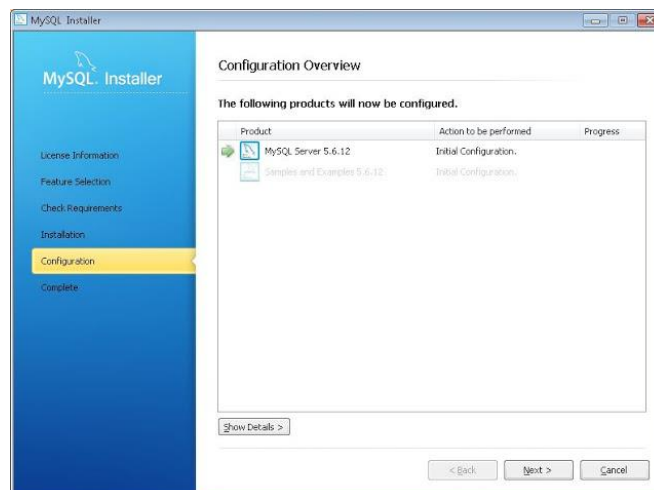




Se iniciará la instalación indicará los productos que se instalarán, el asistente mostrará el progreso y resultado de instalación de cada uno de ellos. Tras la instalación pulsaremos "Next":



El asistente para instalar MySQL Community Server nos mostrará los productos que requieren de configuración, en concreto MySQL Server 5.6.12. Pulsaremos "Next" para iniciar el asistente de configuración de MySQL Server:

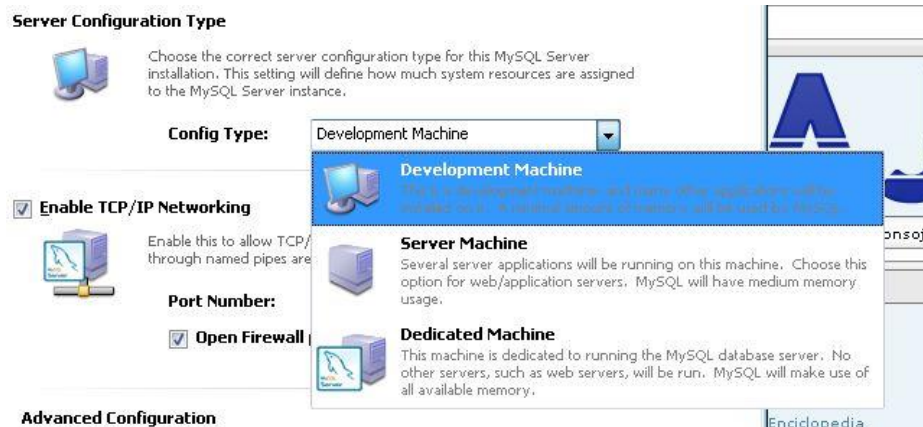


Indicaremos el tipo de configuración para el servidor de MySQL Server en el desplegable "Config Type":

En nuestro caso servirá de servidor de base de datos para pruebas y desarrollo por lo que seleccionaremos "Development Machine":



- ✓ **Development Machine:** para desarrolladores, MySQL Server consumirá pocos recursos y admitirá pocas conexiones concurrentes. Este tipo de instalación es útil para pruebas de desarrollo y test de la base de datos, pero no recomendable para un servidor de producción.

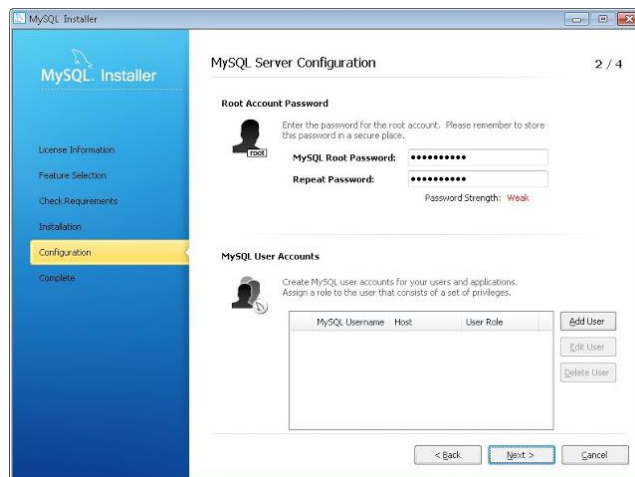


Como el servidor de MySQL Server será accedido desde otros equipos de la red e incluso desde Internet marcaremos "Enable TCP/IP Networking", indicaremos el número de puerto por el que MySQL Server admitirá las conexiones externas al servidor (por defecto el 3306) y si queremos que el asistente abra el puerto correspondiente en el cortafuegos de Windows 7 marcaremos "Open firewall port for network access". Marcaremos también "Show Advanced Options" para poder modificar las opciones avanzadas:

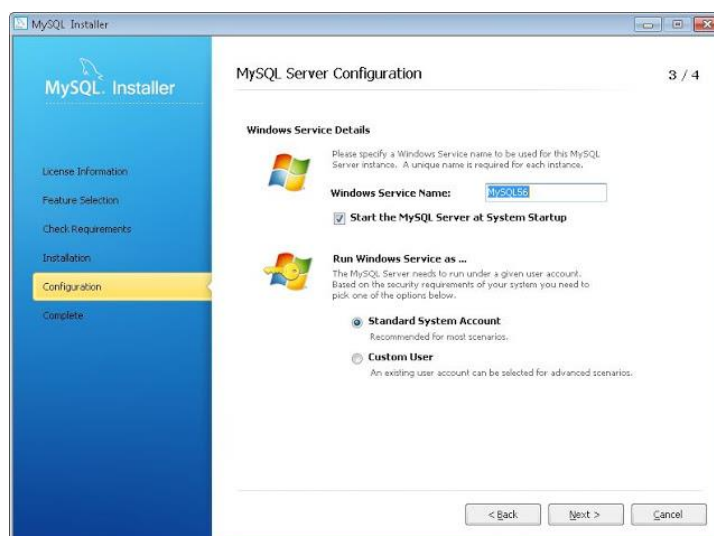




Introduciremos la contraseña para el super usuario root que tendrá permisos de administración sobre el servidor. Desde esta ventana también podremos crear usuarios de MySQL Adicionales pulsando en Add User":

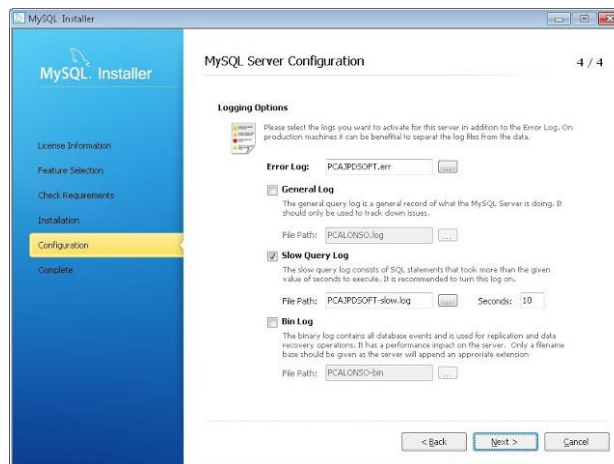


Indicaremos el nombre para el servicio de Windows que creará el asistente. Dicho servicio será el encargado de iniciar automáticamente en el arranque del equipo el servidor de MySQL Server para que esté disponible para recibir conexiones. Para lo que marcaremos "Start the MySQL Server at System Startup". Podremos indicar también el tipo de cuenta de usuario del sistema operativo con el que se iniciará el servicio (cuenta del sistema que suele ser la recomendada "Standard System Account" o bien una específica marcando "Custom User"):





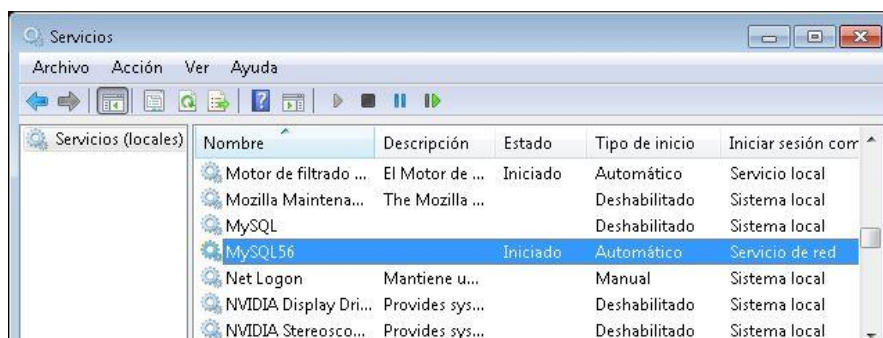
A continuación estableceremos la ubicación y nombre de los ficheros de log de MySQL (error log, general log, slow query log y bin log):



El asistente nos indicará los productos que serán configurados, pulsaremos "Next":

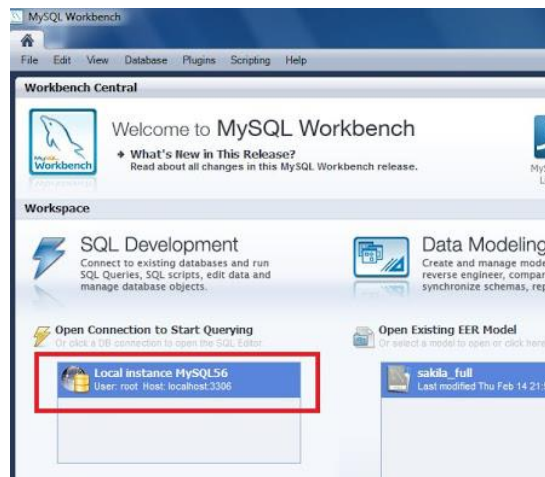
Iniciará la configuración de MySQL Server con los valores establecidos anteriormente, pulsaremos "Next", finalmente nos indicará que la instalación de MySQL Community Server se ha completado. Pulsaremos "Finish":

Ahora podremos comprobar en los "Servicios" de Windows 7 que se ha creado el servicio "MySQL56", para ello desde el botón inicio escribiremos "services.msc" y pulsaremos INTRO, en la ventana de Servicios buscaremos "MySQL56" y comprobaremos que está en Estado "Iniciado":



5.1.1.1 Crear base de datos patronatobd en MySQL Server 5.6

Desde MySQL Workbench, en "Open Connection to Start Querying" ejecutaremos el acceso al servidor de MySQL Server 5.6 de que dispongamos.



Introduciremos la contraseña para el usuario elegido en la conexión a MySQL Server:



Para crear nuestra base de datos llamada patronatobd para nuestro “SISTEMA WEB PARA INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN DEL PATRONATO DE PROMOCIÓN DE SOCIAL RUMIÑAHUI” pulsaremos en el botón "Create a new SQL" y copiar la información que contiene el Dump , pulsaremos ejecutar consulta. Si la consulta SQL es correcta pulsaremos se nos creara la base de datos patronatobd.



```

1 * CREATE DATABASE IF NOT EXISTS 'patronatobd' /*!40100 DEFAULT CHARACTER SET utf8 */;
2 * USE 'patronatobd';
3 -- MySQL dump 10.13 Distrib 5.6.11, for Win64 (x86_64)
4 --
5 -- Host: localhost    Database: patronatobd
6 --
7 -- Server version: 5.6.11
8
9 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
10 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
11 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
12 /*!40101 SET NAMES utf8 */;
13 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
14 /*!40103 SET TIME_ZONE='+00:00' */;
15 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
16 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;

```

Time	Action	Message	Duration / Fetch
272 10.06.33	/!40103 SET TIME_ZONE=@@TIME_ZONE */	0 row(s) affected	0.000 sec
273 10.06.33	/!40101 SET SQL_MODE=@@OLD_SQL_MODE */	0 row(s) affected	0.000 sec
274 10.06.33	/!40014 SET FOREIGN_KEY_CHECKS=@@OLD_FOREIGN_KEY_CHECKS */	0 row(s) affected	0.000 sec
275 10.06.33	/!40014 SET UNIQUE_CHECKS=@@OLD_UNIQUE_CHECKS */	0 row(s) affected	0.000 sec
276 10.06.33	/!40101 SET CHARACTER_SET_CLIENT=@@OLD_CHARACTER_SET_CLIENT */	0 row(s) affected	0.000 sec
277 10.06.33	/!40101 SET CHARACTER_SET_RESULTS=@@OLD_CHARACTER_SET_RESULTS */	0 row(s) affected	0.000 sec
278 10.06.33	/!40101 SET COLLATION_CONNECTION=@@OLD_COLLATION_CONNECTION */	0 row(s) affected	0.000 sec
279 10.06.33	/!40111 SET SQL_NOTES=@@OLD_SQL_NOTES */	0 row(s) affected	0.000 sec

5.1.2. Instalación de jboss-6.1.0.Final en CentOS 6

Para la instalación debemos seguir los siguientes pasos.

✓ Instalación JDK 1.6.0

Para la instalar el JDK se debe ejecute el siguiente comando, **\$ su-c "yum install java-1.7.0- openjdk-devel "**, para confirmar que la versión correcta del JDK está en la ruta de clases: **\$ java-version**.

✓ Descargar JBoss y el procedimiento de instalación

El siguiente paso es descargar la versión adecuada de JBoss 6.

Para eso debemos ir a la página **<http://www.jboss.org/jbossas/downloads/>**

Una vez completada la descarga, ejecutamos el siguiente comando, **\$Unzip jboss-6.1.0.Final.zip-d / java /**. Alternativamente, cualquier directorio puede ser elegido para la instalación de JBoss 6.



CAPÍTULO 6

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

1. Los sistemas de información son de suma importancia para las instituciones porque son una herramienta para el apoyo en la toma de decisiones a nivel operativo.
2. Para el proceso de desarrollo de la solución la parte primordial fue la comprensión y buen entendimiento de la lógica del negocio de la organización, la identificación de forma correcta de todos los requerimientos solicitados.
3. El SISTEMA WEB PARA INSCRIPCIÓN A LOS CURSOS DE CAPACITACIÓN DEL PATRONATO DE PROMOCIÓN SOCIAL DE RUMIÑAHUI , fue desarrollado para mejorar y disminuir tiempos en el proceso de Inscripción y Matriculación a los cursos de capacitación, cumpliendo en su totalidad con los requerimientos establecidos por el Coordinador del Patronato de promoción Social de Rumiñahui.
4. Con el desarrollo del sistema, se proporcionó una herramienta útil para el usuario final, se automatizaron las distintas fases que forman parte del proceso, que anteriormente se llevaban de forma manual y en documentos de Excel lo que permite que sea un proceso eficaz y que genere los resultados previstos en corto tiempo.
5. El sistema es una aplicación web muy ligera, basado en el lenguaje de programación Java con tecnología JSF, usando una interfaz con un diseño amigable, con la finalidad que el usuario interactúe con el sistema sin dificultad.
6. El prototipo fue desarrollado mediante el modelo en espiral, para la ingeniería de software es en la actualidad el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque evolutivo, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel del modelo en espiral.



6.2. Recomendaciones

Las recomendaciones generadas luego de la realización de éste proyecto son las siguientes:

1. Se recomienda la creación de prototipos y la exposición de los mismos con el cliente en forma recursiva y desde las primeras etapas del desarrollo del sistema, para que sea el cliente quien juzgue desde la apariencia de la interfaz de la aplicación hasta el último detalle de la funcionalidad de la solución, incluso en los formatos para la emisión de reportes. De esta manera garantizamos la completa satisfacción del cliente por el producto final entregado.
2. Cada reunión con las personas responsables de la coordinación de los cursos de capacitación debe ser aprovechada al máximo, ya que éste conoce de lógica del negocio pero puede no estar muy claro en la manera de atacar el problema visto desde el punto de vista tecnológico, es ahí, donde es necesaria nuestra intervención como consultores informáticos, para brindarle verdaderas soluciones a sus necesidades, y así evitar que lo que se espera sea una solución se convierta en un problema a mediano o largo plazo tanto para los desarrolladores como para el cliente.
3. Todas las evaluaciones que el cliente realice a los prototipos entregados, deben ser aprovechadas como capacitaciones del funcionamiento del producto final, esto sirve para familiarizar a los actores que interactuarán con el sistema y facilitará la aceptación y comprensión del usuario sobre la herramienta desarrollada.
4. Recordar que el uso de herramientas libres se basa, en cuatro libertades de los usuarios del software:
 - ✓ La libertad de usar el programa con cualquier propósito.
 - ✓ La libertad de estudiar el funcionamiento del programa y adaptarlo a las necesidades.
 - ✓ La libertad de distribuir copias, con lo cual se puede ayudar a otros.



- ✓ La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie
- 5. El administrador del sistema debe ser un profesional en el área de informática o carrera a fines, él mismo que debe tener el conocimiento adecuado de las herramientas, ya que es el único que tiene el acceso total.
- 6. Se recomienda llevar un control adecuado de versiones utilizando un software de control de versiones.
- 7. Aplicar una política de respaldos efectiva de las bases de datos para evitar inconvenientes con posibles fallos en la infraestructura de hardware donde se aloja la solución.



BIBLIOGRAFÍA

1. [http://www.monografias.com/trabajos28/sistema-INSCRIPCIÓN/sistema-INSCRIPCIÓN 2.shtml#ixzz2kdXer9Pd](http://www.monografias.com/trabajos28/sistema-INSCRIPCIÓN/sistema-INSCRIPCIÓN%202.shtml#ixzz2kdXer9Pd)
2. <http://adolfoaraujo.wordpress.com/2012/09/14/apt07-trianguulo-del-proyecto-y-su-calidad/>
3. [http://www.monografias.com/trabajos70/disenio-sistema-informacion-proceso-INSCRIPCIÓN/disenio-sistema-informacion-proceso-INSCRIPCIÓN 2.shtml#ixzz2mouWoinK](http://www.monografias.com/trabajos70/disenio-sistema-informacion-proceso-INSCRIPCIÓN/disenio-sistema-informacion-proceso-INSCRIPCIÓN%202.shtml#ixzz2mouWoinK)
4. <https://sites.google.com/site/ivangarciasanchez90/objetivos/gestion-tema-9/4o>
5. http://laurel.datsi.fi.upm.es/~ssoo/DAW/Trabajos/2003-2004/Septiembre/19/Contenido_archivos/resource1/r1conten2-1.htm
6. <http://asteriscus.com/presentacion/presentacion%20A71922.pdf>
7. https://docs.google.com/document/preview?hgd=1&id=1ACvEZgyTjKj_6tqPwbi-gfG_BZr8H7A_fatsUWYpBeA&pli=1#
8. http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf
9. http://es.wikipedia.org/wiki/JavaServer_Faces
10. <http://es.scribd.com/doc/28079982/Curso-de-JSF-2-0-con-Hibernate-3>
11. <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/eclipse.html>
12. <http://es.wikipedia.org/wiki/JBoss>
13. <http://www.muycomputerpro.com/2011/07/28/red-hat-jboss-application-server-7/>
14. <http://martinezjavier.blogspot.com/2010/02/jsf-20-backing-bean-managedbean-o-named.html>



GLOSARIO

A

Aplicación: Problema o conjunto de problemas para los que se diseña un programa en un lenguaje específico mediante un ordenador.

B

Backup (Copia de Seguridad): Backup es el término inglés que se usa generalmente para hablar de copias de seguridad. Es muy recomendable hacer regularmente respaldos de los datos más importantes, entre ellos las configuraciones de los programas.

Base de datos: Repositorio en el cual se guarda información de forma Organizada que luego será leída, actualizada o eliminada según sean las necesidades.

Browser: Aplicación para visualizar documentos WWW y navegar por el espacio Internet. En su forma más básica son aplicaciones hipertexto que facilitan la navegación por los servidores de información Internet; cuentan con funcionalidades plenamente multimedia y permiten indistintamente la navegación por servidores WWW, FTP, Gopher, el acceso a grupos de noticias, la gestión del correo electrónico, etc.

C

Caso de Uso: Secuencia de acciones que el sistema realiza, la cual proporciona un resultado de valor observable.

E

Eclipse IDE: IDE que facilita el proceso de escribir código en el lenguaje de programación JAVA, entre otros.

EJB: Enterprise Java Beans, permite la implementación de los servicios de negocio dentro de una aplicación con arquitectura JEE.

F

FTP.- (siglas en inglés de File Transfer Protocol, Protocolo de Transferencia de



Archivos) en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor.

H

HTML.- Es el lenguaje que se emplea para el desarrollo de páginas de internet. Este lenguaje está constituido de elementos que el navegador interpreta y las despliega en la pantalla de acuerdo a su objetivo.

HTTP.- Protocolo de transferencia de hipertexto (HyperText transfer Protocol). Controla el intercambio de documentos de hipertexto entre servidores y clientes.

I

Interfaz.- Una interfaz de software es la parte de una aplicación que el usuario ve y con la cual interactúa. Está relacionada con la subyacente estructura, la arquitectura, y el código que hace el trabajo del software, pero no se confunde con ellos. La interfaz incluye las pantallas, ventanas, controles, menús, metáforas, la ayuda en línea, la documentación y el entrenamiento. Cualquier cosa que el usuario ve y con lo cual interactúa es parte de la interfaz.

J

JAVA: Un lenguaje de programación orientado a objetos multiplataforma construido por la empresa SUN Microsystems.

JEE: Abreviatura utilizada para hacer referencia a aplicaciones Empresariales construidas con java, viene del inglés Java Enterprise Edition. JPA: API de persistencia parte de la plataforma JEE.

JSF: Java Server Faces framework parte de la plataforma JEE usado para escribir páginas web.

M

MVC.- Modelo Vista Controlador es un patrón de arquitectura de software que se utiliza mucho en aplicaciones web, separando la en tres capas (modelo,



controlador y vista).

El Modelo se encarga de interactuar con la base de datos y también se ejecuta las reglas de negocio.

El Controlador procesa las peticiones de la página web (vista), y envía estos datos a la capa modelo, para que esta le devuelva la información adecuada para mostrarla en la capa vista.

La vista es el código HTML que se muestra al usuario, con la información proveniente del controlador.

MySQL.- MySQL es la base de datos de software libre más famosa del Mundo, respetada por su velocidad y fiabilidad.

P

PrimeFaces: es un componente para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes ricos que facilitan la creación de las aplicaciones web.

R

RUP/Easy.- El Proceso Racional Unificado (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

S

Sistema de Información: conjunto de sistemas y subsistemas computacionales que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

U

UML.-Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un



estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.



ANEXOS



UNIVERSIDAD CENTRAL DEL ECUADOR
FACULTAD DE INGENIERÍA CIENCIAS FÍSICAS Y MATEMÁTICAS
CARRERA DE INGENIERÍA INFORMÁTICA



ANEXO A: MANUAL DE USUARIO

“SISTEMA WEB PARA INSCRIPCION A LOS CURSOS DE CAPACITACION
DEL PATRONATO DE PROMOCION SOCIAL DE RUMIÑAHUI”



CONTENIDO

INTRODUCCIÓN	2
OBJETIVO	2
DESTINATARIO	2
Simbología	3
1. INGRESO AL SISTEMA	4
1.1 MENÚ PÚBLICO	4
1.1.1 Inicio	4
1.1.2 Autoridades	5
1.1.3 Galería	5
1.1.4 Capacitación	6
1.1.5 Servicios	7
1.1.6 Sistema de Inscripción	8
1.1.6.1 Login	9
1.1.6.1 Ficha Personal	10
1.1.6.2 Inscripción Curso	10
1.2 MENU ADMINISTRADOR	12
1.2.1 Ubicación	12
1.2.2 Cursos	15
1.2.3 Profesión	17
1.2.4 Empleados	20
1.2.5 Contratos	23
1.2.6 Eventos	25
1.2.7 Alumno	29
1.2.8 Matriculación	32
1.2.9 Recaudación	36
1.2.10 Cerrar sesión	37



INTRODUCCIÓN

El sistema web de inscripción y matriculación es un sistema Web programado en la herramienta libre Java JEE, permitiendo a la secretaria del Patronato de Promoción Social tener un fácil y rápido acceso a la información de los estudiantes, eventos, cupos disponibles, cursos, usando un navegador Web.

OBJETIVO

Presentar al usuario final las funcionalidades para el mantenimiento y consulta de la información del sistema web de inscripción y matriculación.

DESTINATARIO

La secretaria del Patronato de Promoción Social encargada del ingreso y modificación de la información. Se requiere conocimiento básico en el uso de navegadores y un conocimiento técnico sobre el tema.

Conocimientos mínimos que deben tener los usuarios del sistema.

- ✓ Conocimientos básicos acerca de programas utilitarios.
- ✓ Conocimientos básicos de navegación Web.
- ✓ Conocimientos básicos de Windows.



A continuación se describe la funcionalidad del sistema web de inscripción y matriculación, la misma que se ha dividido en dos partes, la de administración de la información del Sistema y el proceso de inscripción y matriculación.

Se comenzará describiendo la de administración la cual se detalla en el siguiente apartad

Simbología

Éste proyecto contiene símbolos, para lograr claridad y entendimiento del sistema se detallan a continuación.

- ✓ Banner.-Se presenta en todas las pantallas del sistema como la cabecera.



- ✓ Botón nuevo.- Realiza en todas las pantallas del sistema la acción de limpiar un formulario para ingresar datos.



- ✓ Botón guardar.- Realiza en todas las pantallas del sistema la acción de guardar o actualizar los datos de un formulario.



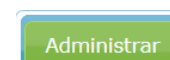
- ✓ Botón editar.- Realiza en todas las pantallas del sistema la acción de una vez seleccionado el dato sube los datos en un formulario para actualizar los campos.



- ✓ Botón eliminar.- Realiza la acción en todas las pantalla de eliminar el dato seleccionado



- ✓ Panel Administrar.- Formulario para ingresar los datos



- ✓ Panel Listar.- Tabla que despliega los datos ingresados



- ✓ Botón exportar datos a excel.- Exporta los datos de la pantalla a formato excel.



- ✓ Botón imprimir.- Imprime reporte en formato pdf.





1. INGRESO AL SISTEMA

Para ingresar al sistema debemos abrir cualquier navegador. Ingresamos la siguiente dirección web del proyecto en el browser <http://localhost:8080/prime/pages/public/principal.jsf>

Una vez ingresada la dirección nos despliega la página principal del Sistema MATSYS (Sistema de Matriculación), el sistema se diseñó un tipo portal con la información importante del Patronato de Promoción Social de Rumiñahui por los que tenemos un menú público y un menú de administrador los cuales los describiré a continuación:



1.1 MENÚ PÚBLICO




1.1.1 Inicio

Esta pantalla despliega la misión, visión y objetivos del Patronato de Promoción Social de Rumiñahui.




Misión



» MISIÓN

Ser una institución líder y referente a nivel nacional, en la gestión social y cultural liderando procesos de competitividad para el mejoramiento integral de la niñez, juventud, tercera edad y personas con capacidades especiales del cantón Rumiñahui, orientado a alcanzar la igualdad de oportunidades como actores del presente y futuro de su propio proyecto de vida


Visión



» VISIÓN

Somos una organización pública, que brinda servicio social a los sectores vulnerables y no vulnerables del cantón Rumiñahui, con capacitación, asistencia en salud integral, apoyando en la creación de microempresas con profesionalismo, calidad, calidez y responsabilidad, con recursos públicos y de autogestión, a fin de alcanzar el buen vivir, mediante el aporte de profesionales y técnicos comprometidos que cristalicen los proyectos sociales, culturales para mejorar el desarrollo integral del ser humano

Objetivos




» OBJETIVOS

Fortalecer el trabajo social del Gobierno Autónomo Descentralizado de Rumiñahui, ejecutando programas de Capacitación, Desarrollo Humano Integral, Desarrollo Cultural y Productivo, a fin de garantizar los derechos de grupos de atención prioritaria con profesionales altamente calificados y una infraestructura física y tecnológica adecuada para su eficiente funcionamiento

1.1.2 Autoridades


Esta pantalla presenta la información de las autoridades que conforman el Patronato de Promoción Social de Rumiñahui.

Presidenta del Patronato del Promocion Social de Rumiñahui



Sra. Ruth Tapia de Jácome, Queridos amigos: Siempre he pensado que para construir una sociedad justa es indispensable la asistencia social a aquellos que más necesitan. Por tanto, desde que tengo el honor de presidir el Patronato de Promoción Social del GADMUR, el centro de mi trabajo se ha vuelto la comunidad, de la que he sido parte toda la vida. Es así que he tomado la decisión de fortalecer la entidad a mi cargo, con autonomía económica y administrativa, para permitir la eficiencia y la eficacia en la ejecución de proyectos sociales que permitan a los ciudadanos ejercer su ciudadanía con igualdad de oportunidades. Miles de personas han sido beneficiadas a través de los diferentes servicios que ofrece el Patronato de Promoción Social en el área de salud integral y en los Programas de Capacitación, de Desarrollo Productivo, de Desarrollo Cultural y de Apoyo Social. Por estas razones el Patronato se ha convertido en el gran brazo ejecutor de la inversión social de la presente Administración Municipal. La solidaridad es el compromiso determinado y firme de trabajar por el bien común. Ese precisamente es el principio de nuestro trabajo

Coordinador Patronato Promocion Social de Rumiñahui



Sr. Andrés Medina, El Patronato propuso a la comunidad un sistema de capacitación que se inicia con el fortalecimiento humano, continúa con la asociatividad, con el sentido comunitario y culmina con el emprendimiento empresarial. Con este proyecto las personas que esperaban ayuda incondicional y permanente de los gobiernos central o seccional para solucionar sus problemas, ahora confían en su capacidad para producir y comercializar los productos que generan.

1.1.3 Galería

Esta pantalla presenta una animación realizada en Flash de los mejores momentos del Patronato de Promoción Social de Rumiñahui en el año 2013.



1.1.4 Capacitación

Esta pantalla es informativa, aquí se despliega información de los cursos de capacitación que se encuentran activos, disponibilidad de cupos, horarios, fecha de inicio y fecha de finalización.

Capacitación					
Cursos	Instructor	Fecha inicio	Fecha Fin	Cupos disponibles	Horario
COMPUTACION, NIVEL I	FEDERICO PATRICIO AVILA JACOME	2014-03-03	2014-04-30	20	LUNES, MIERCOLES Y VIERNES, 8:00 A 10:00
COMPUTACION, NIVEL I	FEDERICO PATRICIO AVILA JACOME	2014-02-03	2014-04-30	20	MARTES Y JUEVES, DE 8:00 A 11:00
COMPUTACION, NIVEL I	FEDERICO PATRICIO AVILA JACOME	2014-03-03	2014-04-30	20	LUNES, MIERCOLES Y VIERNES, 16:00 A 20:00
COMPUTACION, NIVEL I	FEDERICO PATRICIO AVILA JACOME	2014-03-03	2014-04-30	20	SABADOS, 8:00 A 13:00
GUIARRA, NIVEL I	EDISON ALFONSO RAMOS FLORES	2014-02-03	2014-04-30	20	MARTES Y JUEVES, 14:00 A 17:00
CHOCOLATERIA, NIVEL I	BERTHA LUCIA ALDAS MAYANQUER	2014-02-03	2014-04-30	15	LUNES, MIERCOLES Y VIERNES, 11:00 A 13:00
CORTE Y CONFECCION, NIVEL I	MARIA JESUS TELLO VACA	2014-02-03	2014-04-30	15	MARTES Y JUEVES, 8:00 A 11:00
CORTE Y CONFECCION, NIVEL I	MARIA JESUS TELLO VACA	2014-02-03	2014-04-30	15	LUNES Y MIERCOLES, 14:00 A 17:00

- ✓ Botón imprimir.- Para obtener el reporte de los cursos de capacitación en formato pdf dar click en la opción de imprimir, se puede guardar o abrir archivo.





CURSOS DE CAPACITACION 2014

CURSO	HORARIO	DIRIGIDO POR:	FECHA	DISPONIBLE
CASA DE LA NIÑEZ, JUVENTUD Y TERCER EDAD AULA 5				
MASAJES ANTIESTRES	LUNES, MIÉRCOLES Y VIERNES, 15:00 A 17:00	AURA JANNETH MUÑOZ	2014-02-01 A 2014-02-28	25
LA TOLITA, AULA 1				
CHOCOLATERIA	LUNES, MIÉRCOLES Y VIERNES, 11:00 A 13:00	BERTHA LUCIA ALDAS MAYANQUER	2014-02-03 A 2014-04-30	15
GASTRONOMIA	MARTES Y JUEVES, 14:00 A 17:00	PAUKO CESAR VILLACRES CRISANTO	2014-02-03 A 2014-03-31	15
COCINA INTERNACIONAL	LUNES, MIÉRCOLES Y VIERNES, 8:00 A 10:00	PAUKO CESAR VILLACRES CRISANTO	2014-02-01 A 2014-02-28	15
COCINA INTERNACIONAL	SABADOS, 8:00 A 12:00	PAUKO CESAR VILLACRES CRISANTO	2014-02-03 A 2014-03-31	15
PANADERIA, PASTELERIA Y BOCADITOS	LUNES, MIÉRCOLES Y VIERNES, 14:00 A 16:00	VICENTE ERNESTO REYES PAREDES	2014-02-03 A 2014-03-31	15
PANADERIA, PASTELERIA Y BOCADITOS	MARTES Y JUEVES, 8:00 A 11:00	VICENTE ERNESTO REYES PAREDES	2014-02-03 A 2014-03-31	13
PANADERIA, PASTELERIA Y BOCADITOS	SABADOS, 14:00 A 17:00	VICENTE ERNESTO REYES PAREDES	2014-02-01 A 2014-02-28	15

1.1.5 Servicios

Esta pantalla es informativa nos presenta los servicios que brinda el Patronato de Promoción Social de Rumiñahui a la comunidad como:

✓ Capacitación

Capacitación

• CENTRO CULTURAL VILLA CARMEN (Computación básica, Internet, Diseño gráfico, Baile, Guitarra, Dibujo y pintura infantil, Gimnasia para la tercera edad, Teatro) Dirección: Calles Montufar y Espejo





• CENTRO DE CAPACITACIÓN LA TOLITA (Gastronomía, Cocina internacional, Chocolatería ,Panadería ,Pastelería ,Bocaditos ,Arreglos florales ,Belleza ,Tejido ,Bisutería ,Bordados con cintas ,Corte y confección Manualidades) Dirección: Avenida General Enriquez y barrio La Tola

• CAPACITACIÓN EXTERNA (Aeróbicos, Tai chi, Yoga) Dirección: Parque Santa Clara



Ayuda Social

ifñahui Cia.Ltda - 2013

✓ Ayuda social



Capacitación

Ayuda Social

Desarrollo Productivo

• CENTRO "AMIGO GUÍA" - Apoyo y refuerzo escolar a niños y niñas de las escuelas fiscales del cantón Rumiñahui, tareas dirigidas: - Escuela para Padres de Familia de los infantes atendidos con metodologías adecuadas que permitan el desarrollo escolar y mejoren la relación intrafamiliar. Dirección: Barrio San Nicolás Av. General Enriquez. Horario de atención: Lunes a viernes: 8h00 a 17h00 en las oficinas del Patronato Lunes a viernes: 8h30 a 13h00 y de 14h00 a 16h30 en el Centro Amigo Guía.

• CASA DE LA NIÑEZ, JUVENTUD Y TERCERA EDAD (Medicina general, Psicología clínica y educativa, Asistencia legal, Masajes anti estrés, Baloterapia, Gimnasia para la tercera edad) Horario de atención: lunes a viernes de 8h00 a 13h00 y de 14h00 a 17h00 Se puede comunicar al 2867519. La atención es gratuita

Rumiñahui Cia.Ltda - 2013

✓ Desarrollo productivo

Capacitación

Ayuda Social

Desarrollo Productivo

El Patronato propuso a la comunidad un sistema de capacitación que se inicia con el fortalecimiento humano, continúa con la asociatividad, con el sentido comunitario y culmina con el emprendimiento empresarial (producen y comercializan los productos que generan). Capacitación Pecuaria: Crianza de cuyes. Capacitación Agrícola: Siembra de pasto, forraje y hortalizas orgánicas.

Convenios Interinstitucionales: El Patronato de Promoción Social y INIAP, se comprometieron a optimizar la actividad agrícola en el cantón Rumiñahui, a través de la firma de un convenio interinstitucional. El objetivo es ampliar el campo de acción a varios sectores, para impulsar el desarrollo agrícola sostenible y garantizar la seguridad y soberanía alimentaria en este territorio. Con la aplicación de este programa, las entidades involucradas generarán un espacio de "capacitación a capacitadores" en tecnologías de producción limpia de alimentos y contribuirán a la innovación de la tecnología, con las necesidades del agricultor de forma práctica y participativa.

INIAP INSTITUTO NACIONAL AUTÓNOMO DE INVESTIGACIONES AGROPECUARIAS

Rumiñahui Cia.Ltda - 2013

1.1.6 Sistema de Inscripción

Al ingresar nos aparecerá la pantalla de login para ingresar los campos de usuario y contraseña, una vez validado los datos ingresados accedemos al sistema web. Además esta pantalla hace referencia a la seguridad de la aplicación, es decir que existirán diferentes usuarios con diferentes niveles de acceso a secciones específicas de la misma.



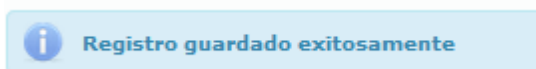
1.1.6.1 Login

- ✓ Para ingresar al sistema de inscripción el alumno debe estar registrado, si no está debe dar click en [Registrarme](#) la opción y le despliega una pantalla para ingresar los datos personales del alumno.

Si el alumno no ingresa los datos requeridos el sistema le despliega un mensaje



Botón guardar.- Si el alumno ingresa los datos correctamente y da click en este botón el sistema le mostrara el mensaje siguiente



- ✓ Una vez registrados en la pantalla de login si ingresamos los datos incorrectos nos presenta un mensaje de error de “Credenciales Incorrectas”



- ✓ Si no ingresa uno de los datos requeridos nos presenta un mensaje de error informándonos del campo requerido.

Una vez el alumno registrado y logeado puede ingresar al sistema de inscripción con su usuario y contraseña antes registrado. Ingresado usuario y contraseña correctos se le despliega una pantalla donde se despliega un menú:

1.1.6.1 Ficha Personal.- Este formulario despliega los datos con los que se registró el alumno, aquí podrá modificar la información que requiera como por ejemplo su

contraseña, ingresar una fotografía.

1.1.6.2 Inscripción Curso.- Este formulario permite al alumno realizar la inscripción a los cursos de capacitación.



1. Se debe seleccionar un curso y automáticamente en el campo Evento se carga todos los eventos que se encuentran disponibles.
2. Seleccionar una opción del evento nos despliega un dialogo con los datos del Evento, damos click en guardar si desea inscribirse en el Evento.

Una vez guardada la inscripción del curso se actualiza nuestra tabla del formulario donde podemos ver todos los cursos que estamos inscritos y además nos despliega un mensaje Inscripción realizada correctamente.



Curso :	Horario :	Costo :
INTERNET, NIVEL 1	SABADOS, 13:00 A 17:00	10.0

Inscripción realizada correctamente

1.2 MENU ADMINISTRADOR

Para ingresar al menú administrador debe tener un usuario y contraseña como administrador, logearse en el sistema.

Si el usuario es administrador ingresara a la siguiente pantalla, donde podrá ingresar, modificar, eliminar información de acuerdo al menú siguiente:



1.2.1 Ubicación

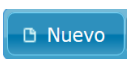

En esta opción se encuentran los datos de los lugares donde se realiza las capacitaciones.

Panel administrar

- ✓ Nombre.- Se debe ingresar el nombre del Centro de capacitación y el aula.
- ✓ Dirección.- Se debe ingresar la dirección del Centro de capacitación.
- ✓ Teléfono.- Se debe ingresar el teléfono del Centro de capacitación.




Botones:

- ✓  **Nuevo** Resetea el formulario, permite ingresar datos en los campos.
- ✓  **Guardar** Permite guardar los datos ingresados

Si no se ingresa datos requeridos el sistema nos despliega el siguiente mensaje

Una vez ingresado los datos correctos en el formulario damos click en guardar el sistema despliega el siguiente mensaje

 Registro guardado exitosamente

Si se ingresa un registro que ya existe le despliega el siguiente mensaje y deberá revisar la información que está ingresando.





 Error al guardar, Ya existe un registro con esos datos

Panel Listar.- Tabla que contiene todos los datos ingresados, cada campo tiene un filtro para buscar. Además se encuentra la opción de exportar a excel todos los datos presentes en la pantalla.

Direccion	Ubicacion	Telefono
SANGOQUI	LA TOLITA,AULA 1	23336321 EXT 102
MONTUFAR Y ESPEJO	VILLA DEL CARMEN SALA DE COMPUTACION	23336321 EXT 101
SANGOQUI	PARQUE SANTA CLARA	23336321 EXT 103
SAN RAFAEL	CASA DE LA JUÑEZ, JUVENTUD Y TERCER EDAD AULA 5	23336321 EXT 104
SANGOQUI	GYMNASIO	23336321 EXT 105
SANGOQUI	FAJARDO	23336321 EXT 106
SANGOQUI	POLIDEPORTIVO	23336321 EXT 107
SANGOQUI	LA TOLITA,AULA 2	23336321 EXT 108
SANGOQUI	LA TOLITA,AULA 3	23336321 EXT 109
MONTUFAR Y ESPEJO	VILLA DEL CARMEN SALON MULTIPLE	23336321 EXT 101



Botones:

- ✓  Permite guardar los datos actualizados.
- ✓  Permite editar la información del registro seleccionado.
- ✓  Permite eliminar el registro seleccionado.
- ✓  Permite exportar los registros de la pantalla a excel.

Si el usuario no selecciona un registro le despliega el siguiente mensaje de alerta

 Se debe seleccionar un ítem

Una vez seleccionado el registro se cargan los datos en el panel de administrar en donde se puede modificar la información y seguido dar click guardar, si los datos son correctos despliega el siguiente mensaje.

 Registro actualizado exitosamente

Si los datos no son correctos despliega el siguiente mensaje

 Error al guardar, Ya existe un registro con esos datos

Para eliminar un registro se debe seleccionar un registro y dar click en eliminar, seguido nos presenta un mensaje de confirmación:

- ✓ Si.- Elimina el registro
- ✓ No.- Cancelo eliminar registro




Si se elimina el registro despliega el siguiente mensaje

 Registro eliminado exitosamente

Si el registro esta siendo utilizado en el sistema despliega el siguiente mensaje



 Error al editar, No se puede borrar Campo esta siendo utilizado



1.2.2 Cursos

En esta opción se encuentran la información de los Cursos de Capacitación para crear un evento.

Panel administrar, se debe ingresar:

- ✓ Nombre.- Se debe ingresar el nombre del curso.
- ✓ Ubicación.- Se debe ingresar la ubicación donde se dictara el curso.
- ✓ Nivel.- Se debe ingresar el nivel del curso.
- ✓ Temario.- Se debe ingresar una descripción breve de los temas que se dictaran en los cursos de capacitación.


Botones:

- ✓  Nuevo Resetea el formulario, permite ingresar datos en los campos.
- ✓  Guardar Permite guardar los datos ingresados

Si no se ingresa datos requeridos el sistema nos despliega el siguiente mensaje



Una vez ingresado los datos correctos en el formulario damos click en guardar el sistema despliega el siguiente mensaje

 Registro guardado exitosamente

Si se ingresa un registro que ya existe le despliega el siguiente mensaje y deberá revisar la información que está ingresando.





 Error al guardar, Ya existe un registro con esos datos

Panel Listar.- Tabla que contiene todos los datos ingresados, cada campo tiene un filtro para buscar. Además se encuentra la opción de exportar a excel todos los datos presentes en la pantalla.



Curso	Lugar	Temario
AEROBICOS/ NIVEL 5	PARQUE SANTA CLARA	BAILOTERPAI
YOGA SOLAR/ NIVEL	PARQUE SANTA CLARA	EJERCICIOS BASICOS
COMPUTACION/ NIVEL	VILLA DEL CARMEN SALA DE COMPUTACION	COMPUTACION BASICA
GUIARRA/ NIVEL	VILLA DEL CARMEN SALON MULTIPLE	CONCEPTOS BASICOS
TEATRO/ NIVEL	VILLA DEL CARMEN SALA DE COMPUTACION	BASICOS
CHOCOLATERIA/ NIVEL	LA TOLITA,AULA 1	BASICOS
CORTE Y CONFECCION/ NIVEL	LA TOLITA,AULA 3	BASICOS
INTERNET/ NIVEL 2	VILLA DEL CARMEN SALA DE COMPUTACION	REDES SOCIALES, NAVEGADORES

Botones:

- ✓  Permite guardar los datos actualizados.
- ✓  Permite editar la información del registro seleccionado.
- ✓  Permite eliminar el registro seleccionado.
- ✓  Permite exportar los registros de la pantalla a excel.

Si el usuario no selecciona un registro le despliega el siguiente mensaje de alerta

 Se debe seleccionar un item

Una vez seleccionado el registro se cargan los datos en el panel de administrar en donde se puede modificar la información y seguido dar click guardar, si los datos son correctos despliega el siguiente mensaje.

 Registro actualizado exitosamente

Si los datos no son correctos despliega el siguiente mensaje

 Error al guardar, Ya existe un registro con esos datos

Para eliminar un registro se debe seleccionar un registro y dar click en eliminar, seguido nos presenta un mensaje de confirmación:


- ✓ Si.- Elimina el registro
- ✓ No.- Canelo eliminar registro



Si se elimina el registro despliega el siguiente mensaje

 Registro eliminado exitosamente

Si el registro esta siendo utilizado en el sistema despliega el siguiente mensaje

 Error al editar, No se puede borrar Campo esta siendo utilizado

1.2.3 Profesión



En este formulario se encuentran la información de profesiones de los capacitadores y empleados del Patronato de Promoción Social de Rumiñahui

Panel administrar, se debe ingresar:

- ✓ Nombre.- Se debe ingresar el nombre de la profesión.

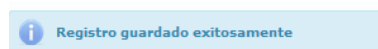


Botones:

- ✓  **Nuevo** Resetea el formulario, permite ingresar datos en los campos.
- ✓  **Guardar** Permite guardar los datos ingresados

Si no se ingresa datos requeridos el sistema nos despliega el siguiente mensaje

Una vez ingresado los datos correctos en el formulario damos click en guardar el sistema despliega el siguiente mensaje



Si se ingresa un registro que ya existe le despliega el siguiente mensaje y deberá revisar la información que está ingresando.





 **Error al guardar, Ya existe un registro con esos datos**

Panel Listar.- Tabla que contiene todos los datos ingresados, cada campo tiene un filtro para buscar. Además se encuentra la opción de exportar a excel todos los datos presentes en la pantalla.

Profesion
BACHILLER-ESTUDIANTE UNIVERSITARIO
ING. COMERCIAL
ING. HOTELERIA Y TURISMO
INSTRUCTOR ARTES MARCIALES
INSTRUCTORA DE BAILE Y GIMNASIA
INSTRUCTORA DE CORTE Y CONFECCION
LICENCIADO EN ARTES PLASTICAS
MAESTRA ARTESANAL
MAESTRA DE ARTES APLICADAS
MUSICO



Botones:

- ✓  Permite guardar los datos actualizados.
- ✓  Permite editar la información del registro seleccionado.
- ✓  Permite eliminar el registro seleccionado.
- ✓  Permite exportar los registros de la pantalla a excel.

Si el usuario no selecciona un registro le despliega el siguiente mensaje de alerta

 Se debe seleccionar un ítem

Una vez seleccionado el registro se cargan los datos en el panel de administrar en donde se puede modificar la información y seguido dar click guardar, si los datos son correctos despliega el siguiente mensaje.

 Registro actualizado exitosamente

Si los datos no son correctos despliega el siguiente mensaje

 Error al guardar, Ya existe un registro con esos datos

Para eliminar un registro se debe seleccionar un registro y dar click en eliminar, seguido nos presenta un mensaje de confirmación:

- ✓ Si.- Elimina el registro
- ✓ No.- Cancelo eliminar registro




Si se elimina el registro despliega el siguiente mensaje

 Registro eliminado exitosamente

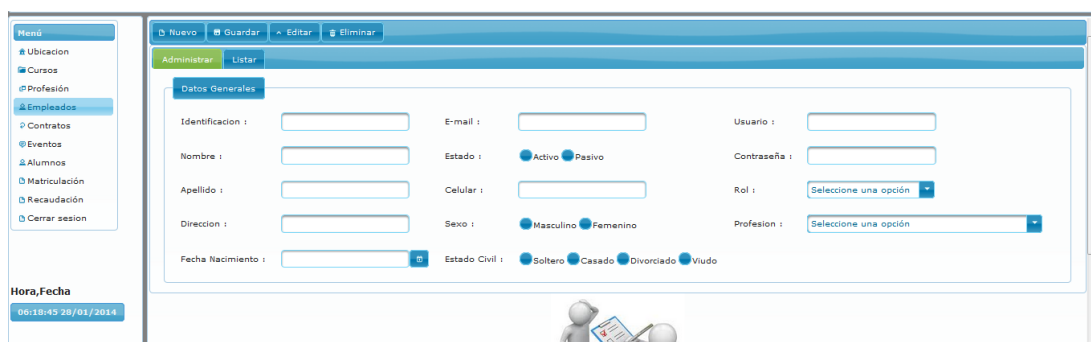
Si el registro esta siendo utilizado en el sistema despliega el siguiente mensaje



 Error al editar, No se puede borrar Campo esta siendo utilizado

1.2.4 Empleados

En este formulario se encuentran la información de los capacitadores y empleados del Patronato de Promoción Social de Rumiñahui.





Panel administrar, se debe ingresar:

- ✓ Identificación.- Se debe ingresar el número de cedula.
- ✓ Nombre.- Se debe ingresar el nombre del empleado.
- ✓ Apellido.- Se debe ingresar el apellido del empleado.
- ✓ Dirección.- Se debe ingresar la dirección del empleado.
- ✓ Teléfono.- Se debe ingresar el teléfono del empleado
- ✓ Email.- Se debe ingresar el correo electrónico del empleado.
- ✓ Estado.- Se debe ingresar el estado del empleado si esta activo o pasivo
- ✓ Celular.- Se debe ingresar el celular del empleado.
- ✓ Sexo.- Se debe ingresar el sexo del empleado.
- ✓ Estado civil.- Se debe ingresar el estado civil del empleado.
- ✓ Usuario.- Se debe ingresar el nombre de usuario del empleado con el cual se ingresara a la aplicación.
- ✓ Contraseña.- Se debe ingresar la contraseña del empleado que será su número de cedula.
- ✓ Rol.- Se debe ingresar el rol del empleado.
- ✓ Profesión.- Se debe ingresar la profesión del empleado.




Botones:

- ✓  **Nuevo** Resetea el formulario, permite ingresar datos en los campos.
- ✓  **Guardar** Permite guardar los datos ingresados

Si no se ingresa datos requeridos el sistema nos despliega el siguiente mensaje

Una vez ingresado los datos correctos en el formulario damos click en guardar el sistema despliega el siguiente mensaje

 **Registro guardado exitosamente**

Si se ingresa un registro que ya existe le despliega el siguiente mensaje y deberá revisar la información que está ingresando.





 **Error al guardar, Ya existe un registro con esos datos**

Panel Listar.- Tabla que contiene todos los datos ingresados, cada campo tiene un filtro para buscar. Además se encuentra la opción de exportar a excel todos los datos presentes en la pantalla.

Cedula Identidad	Nombre	Apellido	Fecha Nacimiento	Email	Celular	Usuario	Profesión
1716893696	MAURICIO DAVID	MORA ARELLANO	1992-07-07	mauriciodma@hotmail.com	0987366860	mauricio	BACHILLER-ESTUDI
170512584-5	FEDERICO PATRICIO	AVILA JACOME	1959-07-03	pateins07@hotmail.com	0987437013	federico	PROGRAMADOR DE
170924486-5	EDISON ALFONSO	RAMOS FLORES	1966-11-24	ediramoss@hotmail.com	0994775894	edison	MUSICO
040075017-0	BERTHA LUCIA	ALDAS MAYANQUER	1964-03-22	lucy.alma@hotmail.com	0991831909	bertha	MAESTRA DE ARTES
170033229-7	MARIA JESUS	TELLO VACA	1945-02-15	isedaguirre@yahoo.com	0987369136	maria	INSTRUCTORA DE C
170972827-1	LUIS HERNAN	SUNTAXI SUQUILLO	1967-06-06	lhusuntaxi@hotmail.com	0998089497	luis	TECNOLOGO EN INF
171326991-6	ISMAEL EFRAIN	MALDONADO VILLARROEL	1976-11-08	ismael2808@hotmail.com	0987473332	ismael	INSTRUCTOR ARTE
171111861-0	LUCI AMAZONAS	PASTRANO ESCOBAR	1967-02-12	luci_5657@hotmail.com	0959713171	luci	MAESTRA ARTESAN



Botones:

- ✓  Permite guardar los datos actualizados.
- ✓  Permite editar la información del registro seleccionado.
- ✓  Permite eliminar el registro seleccionado.
- ✓  Permite exportar los registros de la pantalla a excel.

Si el usuario no selecciona un registro le despliega el siguiente mensaje de alerta

 Se debe seleccionar un ítem

Una vez seleccionado el registro se cargan los datos en el panel de administrar en donde se puede modificar la información y seguido dar click guardar, si los datos son correctos despliega el siguiente mensaje.

 Registro actualizado exitosamente

Si los datos no son correctos despliega el siguiente mensaje

 Error al guardar, Ya existe un registro con esos datos

Para eliminar un registro se debe seleccionar un registro y dar click en eliminar, seguido nos presenta un mensaje de confirmación:

- ✓ Si.- Elimina el registro
- ✓ No.- Cancelo eliminar registro



Si se elimina el registro despliega el siguiente mensaje

 Registro eliminado exitosamente

Si el registro esta siendo utilizado en el sistema despliega el siguiente mensaje



 Error al editar, No se puede borrar Campo esta siendo utilizado



1.2.5 Contratos

Formulario que permite ingresar los datos de los contratos de los capacitadores

Panel administrar, se debe ingresar:

- ✓ Instructor.- Se debe seleccionar el instructor.
- ✓ Fecha inicio.- Se debe ingresar la fecha de inicio del contrato del instructor.
- ✓ Fecha fin.- Se debe ingresar la fecha de finalización del contrato del instructor.
- ✓ Sueldo.- Se debe ingresar el sueldo del empleado.

Botones:

- ✓  Nuevo Resetea el formulario, permite ingresar datos en los campos.
- ✓  Guardar Permite guardar los datos ingresados

Si no se ingresa datos requeridos el sistema nos despliega el siguiente mensaje

Una vez ingresado los datos correctos en el formulario damos click en guardar el sistema despliega el siguiente mensaje



Registro guardado exitosamente

Si se ingresa un registro que ya existe le despliega el siguiente mensaje y deberá revisar la información que está ingresando.

Error al guardar, Ya existe un registro con esos datos

Panel Listar.- Tabla que contiene todos los datos ingresados, cada campo tiene un filtro para buscar. Además se encuentra la opción de exportar a excel todos los datos presentes en la pantalla.

Nuevo

Guardar

Editar


Eliminar

Administrar

Listar

<div><div>«</div><div>«</div><div>1</div><div>»</div><div>»</div></div>		
Empleado	Fecha de Contrato	Sueldo
ALDAS MAYANQUER BERTHA LUCIA	2014-02-01 AL 2014-06-30	495.0
ALENCASTRO NUÑEZ ALBA DEL PILAR	2014-02-01 AL 2014-12-31	495.0
<div><div>«</div><div>«</div><div>1</div><div>»</div><div>»</div></div>		

Exportar Datos



Botones:

- ✓ Permite guardar los datos actualizados.
- ✓ Permite editar la información del registro seleccionado.
- ✓ Permite eliminar el registro seleccionado.
- ✓ Permite exportar los registros de la pantalla a excel.

Si el usuario no selecciona un registro le despliega el siguiente mensaje de alerta

Se debe seleccionar un item

Una vez seleccionado el registro se cargan los datos en el panel de administrar en donde se puede modificar la información y seguido dar click guardar, si los datos son correctos despliega el siguiente mensaje.

Registro actualizado exitosamente

Si los datos no son correctos despliega el siguiente mensaje



 Error al guardar, Ya existe un registro con esos datos

Para eliminar un registro se debe seleccionar un registro y dar click en eliminar, seguido nos presenta un mensaje de confirmación:

- ✓ Si.- Elimina el registro
- ✓ No.- Cancelo eliminar registro



Si se elimina el registro despliega el siguiente mensaje

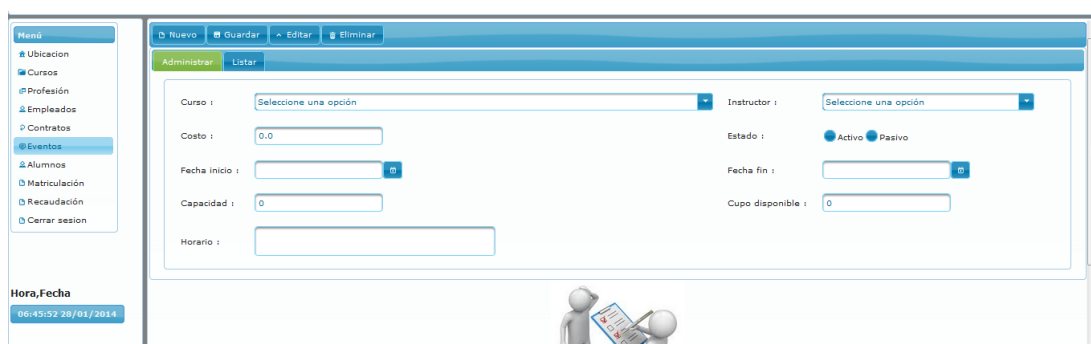
 Registro eliminado exitosamente

Si el registro está siendo utilizado en el sistema despliega el siguiente mensaje

 Error al editar, No se puede borrar Campo esta siendo utilizado

1.2.6 Eventos

Formulario que permite ingresar los datos de los eventos que se realizan en el Patronato de Promoción Social.





Panel administrar, se debe ingresar:

- ✓ Curso.- Se debe seleccionar el curso.
- ✓ Instructor.- Se debe seleccionar el instructor.
- ✓ Costo.- Se debe ingresar el costo.

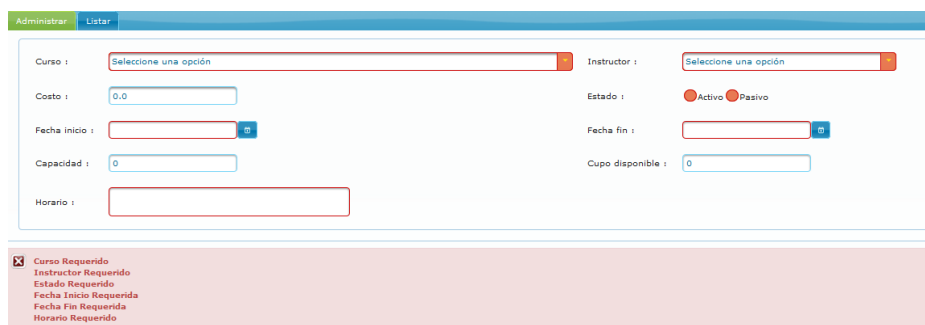


- ✓ Estado.- Se debe ingresar el estado.
- ✓ Fecha inicio.- Se debe ingresar la fecha de inicio del evento.
- ✓ Fecha fin.- Se debe ingresar la fecha de finalización del contrato del evento.
- ✓ Capacidad.- Se debe ingresar la capacidad del evento.
- ✓ Cupos disponibles.- Este campo no se ingresa es automático de acuerdo a las matrículas.
- ✓ Horario.- Se debe ingresar el horario.

Botones:

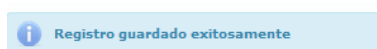
- ✓  Nuevo Resetea el formulario, permite ingresar datos en los campos.
- ✓  Guardar Permite guardar los datos ingresados

Si no se ingresa datos requeridos el sistema nos despliega el siguiente mensaje



The screenshot shows a registration form with the following fields: Curso (dropdown), Instructor (dropdown), Costo (text), Estado (radio buttons for Activo and Pasivo), Fecha inicio (calendar), Fecha fin (calendar), Capacidad (text), Cupo disponible (text), and Horario (text). The Curso, Instructor, Estado, Fecha inicio, Fecha fin, and Cupo disponible fields are highlighted with red borders, indicating they are required. Below the form, a red message box states: 'Curso Requerido, Instructor Requerido, Estado Requerido, Fecha Inicio Requerida, Fecha Fin Requerida, Horario Requerido'.

Una vez ingresado los datos correctos en el formulario damos click en guardar el sistema despliega el siguiente mensaje



Si se ingresa un registro que ya existe le despliega el siguiente mensaje y deberá revisar la información que está ingresando.







Panel Listar.- Tabla que contiene todos los datos ingresados, cada campo tiene un filtro para buscar. Además se encuentra la opción de exportar a excel todos los datos presentes en la pantalla.



Administrar									
Curso	Horario	Capacidad	Cupo Disponible	Fecha Inicio	Fecha Fin	Estado	Lista Alumnos	Inscripción	
DISEÑO GRAFICO,NIVEL, VILLA DEL CARMEN SALA DE COMPUTACION	LUNES, MIERCOLES Y VIERNES, 15:00 A 17:00	25	25	2014-02-01	2014-02-28	A		BERTHA LUCIA ALDAS	
INTERNET,NIVEL, VILLA DEL CARMEN SALA DE COMPUTACION	SABADOS, 13:00 A 17:00	16	16	2014-02-01	2014-02-28	A		LUIS HERNAN SUÑTA	
GIMNASIA PARA LA TERCERA EDAD,NIVEL, VILLA DEL CARMEN SALON MULTIPLE	LUNES, MIERCOLES Y VIERNES, 9:00 A 10:00	60	60	2014-02-01	2014-02-28	A		EDWIN PADILLA BUEN	
DISEÑOS DE JOYAS,NIVEL, VILLA DEL CARMEN SALON MULTIPLE	MARTES Y JUEVES, 8:00 A 11:00	20	20	2014-02-01	2014-02-28	A		CINTHIA BALLOIA	
GIMNASIA PARA LA TERCERA EDAD,NIVEL, CASA DE LA NIÑEZ, JUVENTUD Y TERCER EDAD AULA 5	MARTES Y JUEVES, 8:30 A 9:30	101	101	2014-02-01	2014-02-28	A		EDWIN PADILLA BUEN	
MASAJES ANTIESTRES,NIVEL, CASA DE LA NIÑEZ, JUVENTUD Y TERCER EDAD AULA 5	LUNES, MIERCOLES Y VIERNES, 15:00 A 17:00	25	25	2014-02-01	2014-02-28	A		AURA JANNETH MUÑOZ	
TECNICAS DE BAILE ALTERNATIVO,NIVEL, POLIDEPORTIVO	LUNES A VIERNES, 7:00 A 8:00	15	15	2014-02-01	2014-02-28	A		ALBA DEL PILAR ALEN	
YOGA SOLAR,NIVEL, PARQUE SANTA CLARA	SABADOS Y DOMINGOS, 7:00 A 9:00	20	20	2014-02-01	2014-02-28	A		FEDERICO PATRICKO	
ARTES MARCIALES,NIVEL, PARQUE SANTA CLARA	SABADOS Y DOMINGOS, 9:00 A 11:00	20	20	2014-02-01	2014-02-28	A		ISMAEL EFRAIN MALDI	
BISUTERIA,NIVEL, LA TOLITA,AULA 3	LUNES, MIERCOLES Y VIERNES, 10:30 A 12:30	10	10	2014-02-01	2014-02-28	A		ALEXANDRA GUERRA	


Botones:

- ✓  **Guardar** Permite guardar los datos actualizados.
- ✓  **Editar** Permite editar la información del registro seleccionado.
- ✓  **Eliminar** Permite eliminar el registro seleccionado.
- ✓  **Exportar Datos** Permite exportar los registros de la pantalla a excel.

Si el usuario no selecciona un registro le despliega el siguiente mensaje de alerta

 Se debe seleccionar un ítem

Una vez seleccionado el registro se cargan los datos en el panel de administrar en donde se puede modificar la información y seguido dar click guardar, si los datos son correctos despliega el siguiente mensaje.

 Registro actualizado exitosamente

Si los datos no son correctos despliega el siguiente mensaje

 Error al guardar, Ya existe un registro con esos datos

Para eliminar un registro se debe seleccionar un registro y dar click en eliminar, seguido nos presenta un mensaje de confirmación:

- ✓ Si.- Elimina el registro
- ✓ No.- Canelo eliminar registro



Si se elimina el registro despliega el siguiente mensaje

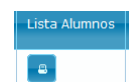
 Registro eliminado exitosamente

Si el registro esta siendo utilizado en el sistema despliega el siguiente mensaje

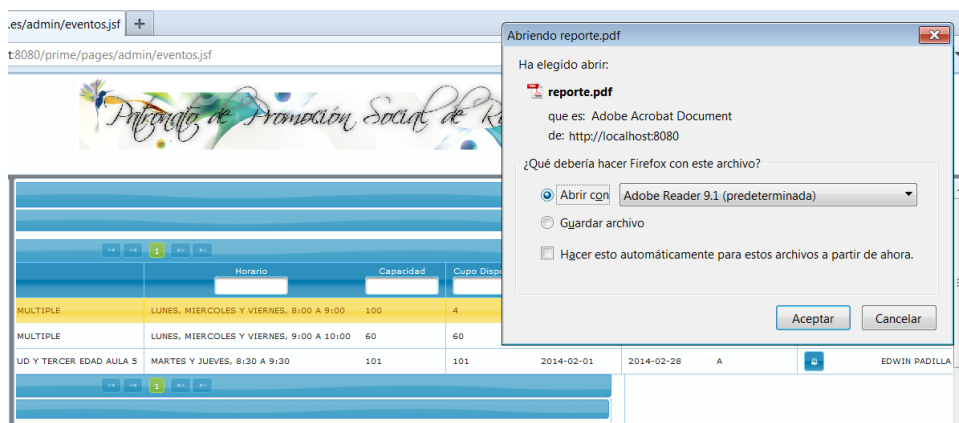
 Error al editar, No se puede borrar Campo esta siendo utilizado

 Error al editar, No se puede borrar Campo esta siendo utilizado

Además en este panel tenemos la opción de sacar un reporte de alumnos matriculados por evento



los





GIMNASIA PARA LA TERCERA EDAD NIVEL

LUGAR: VILLA DEL CARMEN SALON MULTIPLE

CAPACIDAD: 100

DISPONIBLES: 4

HORARIO: LUNES, MIERCOLES Y VIERNES, 8:00 A 9:00

DIRIGIDO POR: EDWIN PADILLA BUENAÑO

FECHA INICIO: 01/02/13 12:00 AM FECHA FIN: 28/02/14 12:00 AM

COMPROBANTE	NOMBRES	CEDULA	TELEFONO	OBSERVACIONES
0005744	VICENTE SIMBAÑA	1700684598	2074004	
0005748	MARIA ISIDORA MARCILLO	1701222000	2870092	
0005749	LUISA CRISTINA FREIRE	1703844678	2087417	
0005750	MANUELA RUALES	1700234386	2332338	
0005751	ELVIA MARTINEZ	1800136424	2860913	
0005752	LUZ AMERICA LONDON	1700875642	2864231	
0005753	DELIA DIAZ	1700047812	2628839	
0005754	CESAR OCTAVIO BARROS	1701646547	6003746	
0005755	MARTHA ZALDUMBIDE	1700144023	6003746	
0005756	HILDA YUGSI	0500090311	3880032	
0005757	TERESA ALVAREZ	060079817	2338817	
0005758	LAURA OLGA DIAZ	1700336553	2870371	

1.2.7 Alumno

Formulario que permite ingresar la información del alumno



Panel administrar, se debe ingresar:

- ✓ Nombre.- Se debe ingresar el nombre del alumno.
- ✓ Apellido.- Se debe ingresar el apellido del alumno.
- ✓ Dirección.- Se debe ingresar la dirección del alumno.
- ✓ Teléfono.- Se debe ingresar el teléfono del alumno
- ✓ Email.- Se debe ingresar el correo electrónico del alumno.
- ✓ Estado.- Se debe ingresar el estado del alumno si esta activo o pasivo

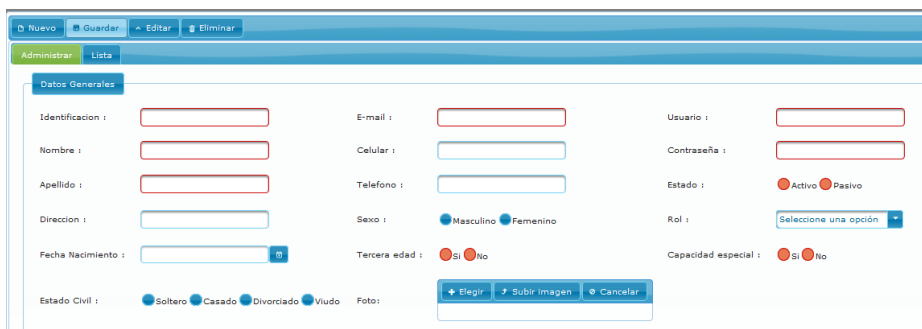


- ✓ Celular.- Se debe ingresar el celular del alumno.
- ✓ Sexo.- Se debe ingresar el sexo del alumno.
- ✓ Estado civil.- Se debe ingresar el estado civil del alumno.
- ✓ Usuario.- Se debe ingresar el nombre de usuario del alumno con el cual se ingresara a la aplicación.
- ✓ Contraseña.- Se debe ingresar la contraseña del alumno que será su número de cedula.
- ✓ Rol.- Se debe ingresar el rol del alumno.
- ✓ Profesión.- Se debe ingresar la profesión del alumno.
- ✓ Tercera edad.- Se debe ingresar si el alumno es de tercera edad ya que con este campo se valida el valor del curso.
- ✓ Capacidad especial.- Se debe ingresar si el alumno tiene alguna discapacidad ya que con este campo se valida el valor de la matrícula del curso.
- ✓ Foto.- Se debe ingresar la foto del alumno.

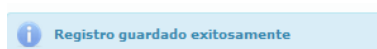
Botones:

- ✓  Resetea el formulario, permite ingresar datos en los campos.
- ✓  Permite guardar los datos ingresados

Si no se ingresa datos requeridos el sistema nos despliega el siguiente mensaje



Una vez ingresado los datos correctos en el formulario damos click en guardar el sistema despliega el siguiente mensaje





Si se ingresa un registro que ya existe le despliega el siguiente mensaje y deberá revisar la información que está ingresando.

Error al guardar, Ya existe un registro con esos datos

Panel Listar.- Tabla que contiene todos los datos ingresados, cada campo tiene un filtro para buscar. Además se encuentra la opción de exportar a excel todos los datos presentes en la pantalla.

Identificación	Nombre	Apellido	Fecha Nacimiento	E-mail	Telefono	Usuario	Imprimir
1710014990	LUIS	ZALDUMBIDE		lzalumbide@hotmail.com		lzalumbide	
171306332-7	PAULINA	PIJAMUNGA		ppijamunga@hotmail.com		ppijamunga	
1716199904	LUZ LORENA	SUVTAKI		lore_sun@gmail.com		lore	
1715187306	MIREYA	GOYES		mireya.goyes1983@gmail.com		mireya	
1712857356	GALO OSWALDO	GAVILANEZ PACHACAMA	1975-02-17	didlospumess@hotmail.com	23804910	galo	
0600295356	ETELVINA	BERRONES		etelvina.berrones@hotmail.com	2332795	etelvina.berrones	
1700684598	VICENTE	SIMBAÑA		vicente.simbana@hotmail.com	2074004	vicente.simbana	
1702290808	LAURA	OÑA		laura.ona@hotmail.com	2074004	laura.ona	

Exportar Datos

Botones:

- ✓ **Guardar** Permite guardar los datos actualizados.
- ✓ **Editar** Permite editar la información del registro seleccionado.
- ✓ **Eliminar** Permite eliminar el registro seleccionado.
- ✓ **Exportar Datos** Permite exportar los registros de la pantalla a excel.

Si el usuario no selecciona un registro le despliega el siguiente mensaje de alerta

Se debe seleccionar un ítem

Una vez seleccionado el registro se cargan los datos en el panel de administrar en donde se puede modificar la información y seguido dar click guardar, si los datos son correctos despliega el siguiente mensaje.

Registro actualizado exitosamente

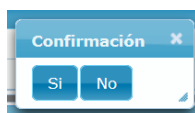
Si los datos no son correctos despliega el siguiente mensaje



 Error al guardar, Ya existe un registro con esos datos

Para eliminar un registro se debe seleccionar un registro y dar click en eliminar, seguido nos presenta un mensaje de confirmación:

- ✓ Si.- Elimina el registro
- ✓ No.- Canelo eliminar registro



Si se elimina el registro despliega el siguiente mensaje

 Registro eliminado exitosamente

Si el registro está siendo utilizado en el sistema despliega el siguiente mensaje

1.2.8 Matriculación

Formulario que permite ingresar los datos de la inscripción y matriculación de un alumno que se realizan en el Patronato de Promoción Social.

Panel administrar.- Permite ingresar la información de las matriculas a los cursos de capacitación del Patronato de Promoción Social, se debe primero verificar que exista el curso, el evento y el alumno sino existe se debe crear.



Panel administrar, se debe ingresar:

- ✓ Curso.- Se debe seleccionar el curso.
- ✓ Evento.- Se debe seleccionar el evento.

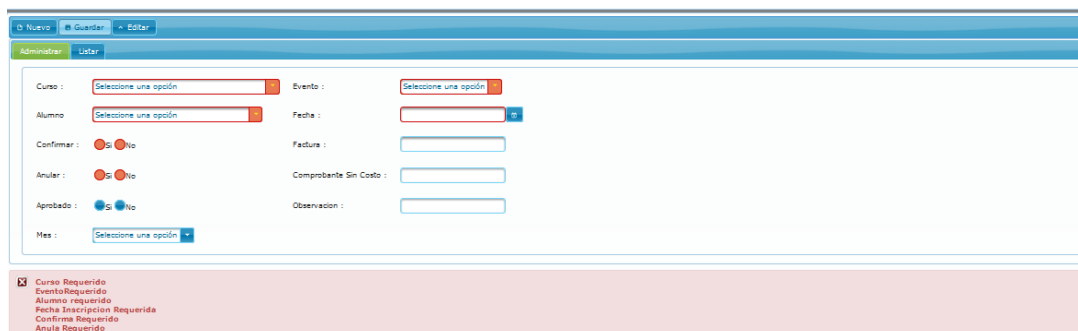


- ✓ Alumno.- Se debe seleccionar el alumno.
- ✓ Fecha.- Se debe ingresar la fecha de inscripción.
- ✓ Confirma.- Se selecciona si confirma para que el alumno pase del estado inscrito a matriculado
- ✓ Anula.- Se debe ingresar si anula la inscripción o matrícula.
- ✓ Aprueba.- Se debe ingresar si el alumno aprueba o no el evento.
- ✓ Mes.- Se debe seleccionar el mes al que se matricula el alumno.
- ✓ Factura.- Se debe ingresar el número de factura.
- ✓ Comprobante sin costo.- Se debe ingresar el número de comprobante sin costo cuando las personas son de la tercera edad y el evento es gimnasia de la tercera edad.
- ✓ Observación.- Se debe ingresar una observación de la matrícula al evento.


Botones:

- ✓  Resetea el formulario, permite ingresar datos en los campos.
- ✓  Permite guardar los datos ingresados

Si no se ingresan datos requeridos el sistema nos despliega el siguiente mensaje



Una vez ingresado los datos correctos en el formulario damos click en guardar el sistema despliega el siguiente mensaje

 Registro guardado exitosamente

Si se ingresa un registro que ya existe le despliega el siguiente mensaje y deberá revisar la información que está ingresando.





 Error al guardar, Ya existe un registro con esos datos




Panel Listar.- Tabla que contiene todos los datos ingresados, cada campo tiene un filtro para buscar. Además se encuentra la opción de exportar a excel todos los datos presentes en la pantalla.

Factura	Comprobante	FACTURA	COMPROBANTE	NOMBRE	CEDULA	VALOR	CURSO	HORARIO	TELEFONO
0987653				BERTHA ALMEIDA	1702132026	5.0	ARREGLOS FLORALES	LUNES Y JUEVES, 8:00 A 11:00	2082008
				FIGUEROA MAGDALENA	1703190494	0.0	INTERNET	SABADOS, 13:00 A 17:00	23804910

Botones:

- ✓  Permite guardar los datos actualizados.
- ✓  Permite editar la información del registro seleccionado.
- ✓  Permite eliminar el registro seleccionado.
- ✓  Permite exportar los registros de la pantalla a excel.

Si el usuario no selecciona un registro le despliega el siguiente mensaje de alerta

 Se debe seleccionar un ítem

Una vez seleccionado el registro se cargan los datos en el panel de administrar en donde se puede modificar la información y seguido dar click guardar, si los datos son correctos despliega el siguiente mensaje.

 Registro actualizado exitosamente

Si los datos no son correctos despliega el siguiente mensaje

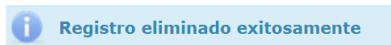
 Error al guardar, Ya existe un registro con esos datos

Para eliminar un registro se debe seleccionar un registro y dar click en eliminar, seguido nos presenta un mensaje de confirmación:

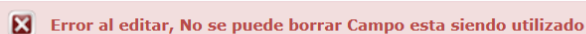
- ✓ Si.- Elimina el registro
- ✓ No.- Canelo eliminar registro



Si se elimina el registro despliega el siguiente mensaje



Si el registro esta siendo utilizado en el sistema despliega el siguiente mensaje



Además en este panel se puede sacar el reporte con los datos de la matrícula para seguidamente imprimir la factura

01/02/2014			
VICENTE SIMBAÑA			
1700684598		2074004	
CONOCOTO			
1	GIMNASIA PARA LA TERCERA EDAD, NIVEL	0,00	0,00
LUNES, MIERCOLES Y VIERNES, 8:00 A 9:00			
			0,00
			0,00
		12	0,00
			0,00

Además en este panel se puede sacar el reporte con los datos de la matrícula para seguidamente imprimir el comprobante sin costo

02/01/2014
ETELVINA BERRONES
0600295356
2332795
DON QUIJOTE

GIMNASIA PARA LA TERCERA EDAD,
NIVEL
LUNES, MIERCOLES Y VIERNES,
8:00 A 9:00
FEBRERO





1.2.9 Recaudación

Permite obtener los reportes de la recaudación de acuerdo a fechas

Para obtener el reporte de recaudación se debe seleccionar una Fecha de inicio y una fecha de fin y dar click en imprimir.



INFORME DE EMISION Y RECAUDACION DE VILLA DEL CARMEN

FECHA: 27 enero 2014

FACTURA	NOMBRES	CEDULA	VT T12%	IVA	TOTAL	CONCEPTO	TELEFONO
0987653	BERTHA ALMEIDA	1702132026	4.46	0.54	5.0	ARREGLOS FLORALES	2082008
0987654	GALO OSWALDO GAVILANEZ	1712857356	8.93	1.07	10.0	ARREGLOS FLORALES	23804910
					15.0		
RESUMEN RECAUDACION							
CURSOS DE CAPACITACION		15.0					
TOTAL		15.0					

ENTREGA: SRA. MIREYA GOYES
RECIBE: BEATRIZ DE PALOMEQUE

INGRESO CURSOS
TESORERIA



1.2.10 Cerrar sesión

Al dar click en esta opción cerramos la sesión salimos del sistema de inscripción y regresamos a la pantalla de logeo.



UNIVERSIDAD CENTRAL DEL ECUADOR
FACULTAD DE INGENIERÍA CIENCIAS FÍSICAS Y MATEMÁTICA
CARRERA DE INGENIERÍA INFORMÁTICA



ANEXO B: MANUAL TECNICO

“SISTEMA WEB PARA INSCRIPCION A LOS CURSOS DE CAPACITACION
DEL PATRONATO DE PROMOCION SOCIAL DE RUMIÑAHUI”



CONTENIDO

INTRODUCCION	2
1. DESARROLLO DE LA APLICACIÓN WEB CON JPA, EJB, JSF	3
1.1 Capa de persistencia.....	3
1.2 Capa de Servicio	4
1.2.1 Clase MatriculaBean.java	5
1.2.2 Paquete AdminEvento.java.....	13
1.3 Capa de presentación	18



INTRODUCCION

En este manual técnico se describe cómo fue desarrollado el “SISTEMA WEB PARA INSCRIPCION A LOS CURSOS DE CAPACITACION DEL PATRONATO DE PROMOCION SOCIAL DE RUMIÑAHUI” y la estructura del mismo.

- ✓ Se utilizó Java Server Faces que es un framework de desarrollo basado en el patrón MVC (Modelo Vista Controlador), porque normaliza y estandariza el desarrollo de aplicaciones web
- ✓ Java Resources aquí están todos los archivos *.java y eclipse se encargará automáticamente de publicar los *.class en donde corresponda
- ✓ Web aquí estarán los archivos *.xhtml, *.jsf, dentro de esta carpeta podemos ver la estructura estándar de JEE para un proyecto Web con la carpeta WEB-INF, el archivo web.xml.



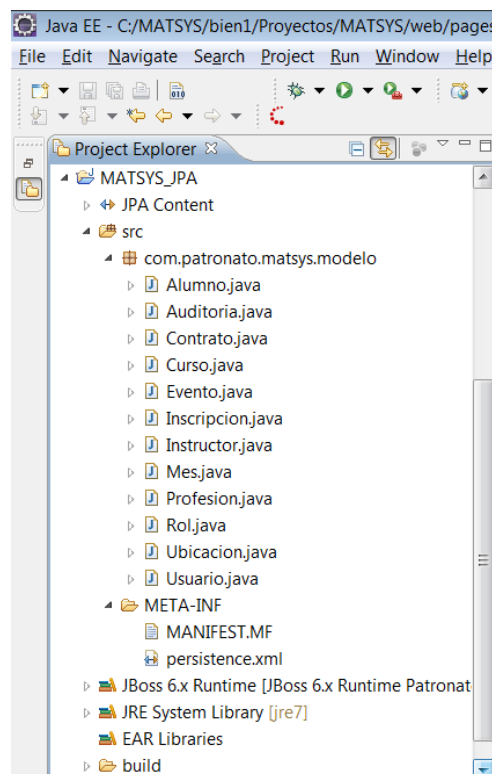
1. DESARROLLO DE LA APLICACIÓN WEB CON JPA, EJB, JSF

1.1 Capa de persistencia

Java Persistence API (JPA).- Proporciona un modelo de persistencia para mapear bases de datos relacionales. JPA que es una abstracción que está por encima de JDBC lo que permite ser independiente de SQL. Todas las clases y anotaciones de esta API se encuentran en el paquete javax.persistence

Los principales componentes de JPA son:

- ✓ Mapeo de base de datos relacionales (ORM). Es el mecanismo para mapear objetos a los datos almacenados en una base de datos relacional.
- ✓ Un API administrador de entidad para realizar operaciones en la base de datos tales como crear, leer, actualizar, eliminar (CRUD).
- ✓ El Java Persistence Query Language (JPQL) que permite recuperar datos con un lenguaje de consultas orientado a objetos.
- ✓ Las transacciones y mecanismos de bloqueo cuando se accedan a los datos concurrentemente, la API Java Transaction (JTA)





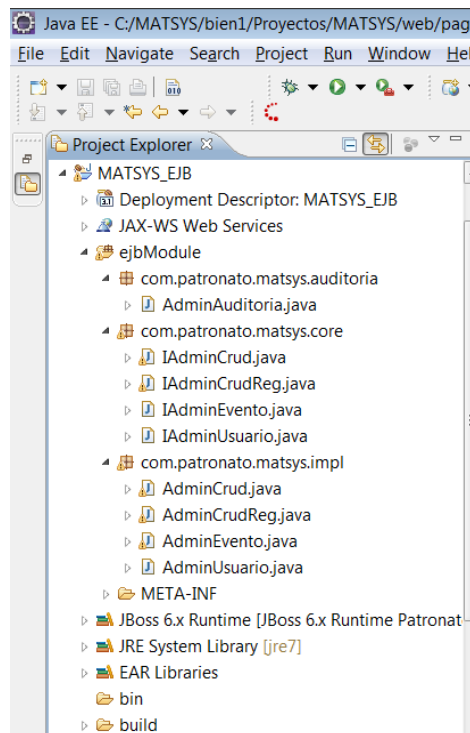
1.2 Capa de Servicio

EJB.- Los JavaBeans empresariales son una tecnología (API) que forma parte del estándar de Java EE

Están diseñados para desarrollo y despliegue de aplicaciones (distribuidas) de negocio basadas en componentes del lado del servidor. Una vez que se desarrolla una aplicación, ésta puede ser desplegada en cualquier servidor que soporte la especificación de EJB.

Con esta tecnología es posible desarrollar aplicaciones empresariales sin tener que crear de nuevo los servicios de transacción, seguridad, persistencia, concurrencia y lo que se pueda necesitar en el proceso de creación de una aplicación; permitiendo a los desarrolladores enfocarse en la implementación de la lógica de negocio.

EJB cuenta con dos componentes de proceso de negocio, los beans de sesión (Session Beans) y los beans dirigidos por mensajes (Message-Driven Beans, MDBs), ambos son desarrollados por una aplicación de negocio e implementados y ejecutados por el Contenedor de EJB





1.2.1 Clase MatriculaBean.java

```
package com.patronato.backingbean;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.annotation.PostConstruct;
import javax.annotation.Resource;
import javax.ejb.EJB;
import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.ViewScoped;
import javax.faces.context.ExternalContext;
import javax.faces.context.FacesContext;
import javax.sql.DataSource;

import org.primefaces.component.tabview.TabView;
import org.primefaces.event.SelectEvent;

import com.patronato.matsys.core.IAdminCrud;
import com.patronato.matsys.core.IAdminEvento;
import com.patronato.matsys.modelo.Alumno;
import com.patronato.matsys.modelo.Curso;
import com.patronato.matsys.modelo.Evento;
import com.patronato.matsys.modelo.Inscripcion;
import com.patronato.matsys.modelo.Mes;
import com.patronato.matsys.modelo.Ubicacion;
import com.patronato.matsys.modelo.Usuario;
import com.patronato.reporte.ApplicationBean;
import com.patronato.reporte.ConfiguracionReporte;
import com.patronato.reporte.ConstanteReporte;
import com.patronato.utils.FacesUtil;

@ManagedBean
@ViewScoped
public class MatriculaBean {
    private List<Inscripcion> inscripciones;
    private List<Curso> cursos;
    private Mes mes;
    private List<Mes> meses;
    private Ubicacion ubicacion;
    private List<Evento> eventos;
    private List<Alumno> alumnos;
    private Inscripcion inscripcion;
    private Inscripcion inscripcionSel;
    private Usuario usuario;
    private String valAnu;
```



```
private String valConf;
private String valAp;

private boolean banEveCur;

@EJB
private IAdminCrud adminCrud;
@EJB
private IAdminEvento adminEvento;

@Resource(mappedName = "java:contasysDS")
private DataSource dsReporte;

public MatriculaBean() {
    this.cursos = new ArrayList<>();
    this.meses= new ArrayList<>();
    this.eventos=new ArrayList<>();
    this.alumnos = new ArrayList<>();
    this.inscripciones=new ArrayList<>();
    this.inscripcion = new Inscripcion();
    this.inscripcion.setEvento(new Evento());
    this.inscripcion.setAlumno(new Alumno());
    this.inscripciones = new ArrayList<>();
}

public Mes getMes() {
    return mes;
}

public void setMes(Mes mes) {
    this.mes = mes;
}

public Ubicacion getUbicacion() {
    return ubicacion;
}

public void setUbicacion(Ubicacion ubicacion) {
    this.ubicacion = ubicacion;
}

public boolean isBanEveCur() {
    return banEveCur;
}

public void setBanEveCur(boolean banEveCur) {
    this.banEveCur = banEveCur;
}
```



```
public String getValAnu() {  
    return valAnu;  
}  
  
public void setValAnu(String valAnu) {  
    this.valAnu = valAnu;  
}  
  
public String getValConf() {  
    return valConf;  
}  
  
public void setValConf(String valConf) {  
    this.valConf = valConf;  
}  
  
public Usuario getUsuario() {  
    return usuario;  
}  
public void setUsuario(Usuario usuario) {  
    this.usuario = usuario;  
}  
public List<Inscripcion> getInscripciones() {  
    return inscripciones;  
}  
public void setInscripciones(List<Inscripcion> inscripciones) {  
    this.inscripciones = inscripciones;  
}  
  
public List<Evento> getEventos() {  
    return eventos;  
}  
  
public void setEventos(List<Evento> eventos) {  
    this.eventos = eventos;  
}  
  
public List<Alumno> getAlumnos() {  
    return alumnos;  
}  
  
public void setAlumnos(List<Alumno> alumnos) {  
    this.alumnos = alumnos;  
}  
  
public Inscripcion getInscripcion() {  
    return inscripcion;  
}  
  
public void setInscripcion(Inscripcion inscripcion) {
```



```

        this.inscripcion = inscripcion;
    }

    public Inscripcion getInscripcionSel() {
        return inscripcionSel;
    }

    public List<Mes> getMeses() {
        return meses;
    }

    public void setMeses(List<Mes> meses) {
        this.meses = meses;
    }

    public void setInscripcionSel(Inscripcion inscripcionSel) {
        this.inscripcionSel = inscripcionSel;
    }

    public String getValAp() {
        return valAp;
    }

    public void setValAp(String valAp) {
        this.valAp = valAp;
    }

    public List<Curso> getCursos() {
        return cursos;
    }

    public void setCursos(List<Curso> cursos) {
        this.cursos = cursos;
    }

    public void cargarEventosCursoIns(){
        try {
            this.eventos
adminEvento.devolverEventosPorCurso(inscripcion.getEvento().getCurso().getIdCursos());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void cargarInscripciones() {
        this.inscripciones.clear();
        List<Object> usuariosTmp
adminCrud.consultarTodo(Inscripcion.class);
        for (Object objTmp : usuariosTmp) {
            this.inscripciones.add((Inscripcion) objTmp);
        }
    }

```



```

    }
    public void guardarActualizar() {
        FacesMessage mensaje = new FacesMessage();
        try {
            mensaje.setSeverity(FacesMessage.SEVERITY_INFO);
            if (inscripcionSel != null) { // Actualizando
                inscripcion.setAnulaIns(valAnu);
                inscripcion.setConfirmaIns(valConf);
                inscripcion.setApruebaIns(valAp);

                mensaje.setSummary(adminEvento.guardarInscripcion(inscripcion));
            } else {
                inscripcion.setAnulaIns(valAnu);
                inscripcion.setConfirmaIns(valConf);
                inscripcion.setApruebaIns(valAp);
                String
                menGua=adminEvento.guardarInscripcion(inscripcion);
                mensaje.setSummary(menGua);
            }
            resetearFormulario();
            inicializar();
        } catch (Exception e) {
            mensaje.setSeverity(FacesMessage.SEVERITY_ERROR);
            mensaje.setSummary("Error al guardar:" + e.getMessage());
        }
        FacesContext.getCurrentInstance().addMessage("Mensaje", mensaje);
    }

    public void editar() {
        FacesMessage mensaje = new FacesMessage();
        try {
            if (inscripcionSel != null) {
                setInscripcion(inscripcionSel);
                cargarEventosCursoIns();
                valAnu = inscripcion.getAnulaIns();
                valConf = inscripcion.getConfirmaIns();
                valAp=inscripcion.getApruebaIns();
                mensaje.setSeverity(FacesMessage.SEVERITY_INFO);
                mensaje.setSummary("Registro cargado
exitosamente");
                TabView tabVieUsu = (TabView)
                FacesContext.getCurrentInstance()

                .getViewRoot().findComponent("frmCon:tabVieUsu");
                tabVieUsu.setActiveIndex(0);
            } else {

                mensaje.setSeverity(FacesMessage.SEVERITY_ERROR);
                mensaje.setSummary("Se debe seleccionar un item");
            }
        }
    }

```



```

    }
    } catch (Exception e) {
        mensaje.setSeverity(FacesMessage.SEVERITY_ERROR);
        mensaje.setSummary("Error al editar:" + e.getMessage());
    }
    FacesContext.getCurrentInstance().addMessage("Mensaje", mensaje);
}

public void eliminar() {
    FacesMessage mensaje = new FacesMessage();
    try {
        if (inscripcionSel != null) {
            mensaje.setSeverity(FacesMessage.SEVERITY_INFO);

            mensaje.setSummary(adminCrud.eliminar(inscripcionSel, "admin"));
        } else {

            mensaje.setSeverity(FacesMessage.SEVERITY_ERROR);
            mensaje.setSummary("Se debe seleccionar un item");
        }
        resetearFormulario();
        inicializar();
    } catch (Exception e) {
        mensaje.setSeverity(FacesMessage.SEVERITY_ERROR);
        mensaje.setSummary("Error al editar:" + e.getMessage());
    }
    FacesContext.getCurrentInstance().addMessage("Mensaje", mensaje);
}

public void seleccionarRegistro(SelectEvent event) {
    setInscripcionSel((Inscripcion) event.getObject());
}

private void cargarListaCursos(){
    this.cursos.clear();
    List<Object> rolesTmp = adminCrud.consultarTodo(Curso.class);
    for (Object objTmp : rolesTmp) {
        this.cursos.add((Curso)objTmp);
    }
}

private void cargarListaMeses(){
    this.meses.clear();
    List<Object> rolesTmp = adminCrud.consultarTodo(Mes.class);
    for (Object objTmp : rolesTmp) {
        this.meses.add((Mes)objTmp);
    }
}

public void generarReporteFactura() {
    try {

```



```

        if (inscripcionSel != null) {
            ExternalContext servletContext = FacesContext
                .getCurrentInstance().getExternalContext();
            String path = servletContext.getRealPath("/");
            Map<String, Object> parameters = new
HashMap<String, Object>();

            String nombreReporte = "FacturaCurso.jasper";
            parameters.put("idIns", inscripcionSel.getIdIns());

            ConfiguracionReporte configuracionReporte = new
ConfiguracionReporte();
            configuracionReporte.setParameters(parameters);

            configuracionReporte.setNombreReportePrincipal(nombreReporte);
            configuracionReporte.setPathReportes("/reportes/");
            configuracionReporte

            .setFormatoExportacion(ConstanteReporte.FORMATO_PDF);

            ApplicationBean appBean = new ApplicationBean();
            try {
                appBean.generarReporte(dsReporte,
configuracionReporte);
            } catch (Exception e) {

                FacesUtil.anadirMensaje(FacesMessage.SEVERITY_ERROR,
                    "No se pudo generar reporte:" +
e.getMessage());
                e.printStackTrace();
            }
        } else {

            FacesUtil.anadirMensaje(FacesMessage.SEVERITY_ERROR,
                "Se debe seleccionar un evento");

        }
    } catch (Exception e) {
        FacesUtil.anadirMensaje(FacesMessage.SEVERITY_ERROR,
            "No se pudo cargar los eventos");
    }

    resetearFormulario();
}

public void generarReporteComprobante() {
    try {
        if (inscripcionSel != null) {
            ExternalContext servletContext = FacesContext

```




```

        .getCurrentInstance().getExternalContext();
        String path = servletContext.getRealPath("/");
        Map<String, Object> parameters = new
HashMap<String, Object>();

        String nombreReporte =
"ComprobanteSinCosto.jasper";

        parameters.put("idIns", inscripcionSel.getIdIns());

        ConfiguracionReporte configuracionReporte = new
ConfiguracionReporte();
        configuracionReporte.setParameters(parameters);

        configuracionReporte.setNombreReportePrincipal(nombreReporte);
        configuracionReporte.setPathReportes("/reportes/");
        configuracionReporte

        .setFormatoExportacion(ConstanteReporte.FORMATO_PDF);

        ApplicationBean appBean = new ApplicationBean();
        try {
            appBean.generarReporte(dsReporte,
configuracionReporte);
        } catch (Exception e) {

            FacesUtil.anadirMensaje(FacesMessage.SEVERITY_ERROR,
                "No se pudo generar reporte:" +
e.getMessage());
            e.printStackTrace();
        }
        } else {

            FacesUtil.anadirMensaje(FacesMessage.SEVERITY_ERROR,
                "Se debe seleccionar un evento");

        }
        } catch (Exception e) {
            FacesUtil.anadirMensaje(FacesMessage.SEVERITY_ERROR,
                "No se pudo cargar los eventos");
        }

        resetearFormulario();
    }

    private void cargarListAlumnos() {
        this.alumnos.clear();
    }

```



```

        List<Alumno> instTmp =
adminEvento.consultarAlumnoOrdenPorNombre("A");
        for (Alumno objTmp : instTmp) {
            this.alumnos.add(objTmp);
        }
    }
    public void resetearFormulario() {
        this.inscripcion = new Inscripcion();
        this.inscripcionSel = null;
        this.valAnu=null;
        this.valAp=null;
        this.valConf=null;
    }

    @PostConstruct
    public void inicializar() {
        cargarInscripciones();
        cargarListaCursos();
        cargarListAlumnos();
        cargarListaMeses();
    }
}

```

1.2.2 Paquete AdminEvento.java

```

package com.patronato.matsys.impl;

import java.util.List;

import javax.ejb.Stateless;
import javax.interceptor.ExcludeClassInterceptors;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

import com.patronato.matsys.core.IAdminEvento;
import com.patronato.matsys.modelo.Alumno;
import com.patronato.matsys.modelo.Evento;
import com.patronato.matsys.modelo.Inscripcion;
import com.patronato.matsys.modelo.Instructor;

@Stateless
public class AdminEvento implements IAdminEvento {
    @PersistenceContext(unitName = "contasysPU")
    private EntityManager em;

    @SuppressWarnings("unchecked")
    @Override

```



```

public List<Evento> devolverEventosPorCurso(int idCur) throws Exception
{
    Query conEveCur = em.createQuery("Select eve from Evento eve
where eve.curso.idCursos =:idCur"
                                     + " and eve.estadoEve =:estadoAct");
    conEveCur.setParameter("idCur", idCur);
    conEveCur.setParameter("estadoAct", "A");
    return conEveCur.getResultList();
}

/*
 * @Override public String guardarInscripcion(Inscripcion inscripcion)
 * throws Exception {
int capEve =
    * inscripcion.getEvento().getCapacidadEve(); if(capEve > 0){
    * em.persist(inscripcion); //inscripcion.getEvento().setCapacidadEve(capEve
    * - 1); inscripcion.getEvento().setCapacidadEve(capEve - 0);
    *
    * em.merge(inscripcion.getEvento()); } return null; }
    */

@Override
public String guardarInscripcion(Inscripcion inscripcion) throws Exception {
    String cadenaInscripcion = "Inscripción ";
    if (inscripcion.getConfirmaIns() != null
        && inscripcion.getConfirmaIns().equals("S")
        && inscripcion.getAlumno().getTerceraEdad() != null
        &&
inscripcion.getAlumno().getTerceraEdad().equals("S")
        &&
inscripcion.getEvento().getCurso().getNombreCurso()
        .equals("GIMNASIA PARA LA TERCERA EDAD"))
    {
        int cupodisEve =
inscripcion.getEvento().getCupoDisponible();
        inscripcion.getEvento().setCupoDisponible(cupodisEve - 1);
        inscripcion.setValor(0);
        em.merge(inscripcion);
        em.merge(inscripcion.getEvento());
        cadenaInscripcion = "Matriculación ";
    } else
    if (inscripcion.getConfirmaIns() != null
        && inscripcion.getConfirmaIns().equals("S")
        && inscripcion.getAlumno().getTerceraEdad() != null
        &&
inscripcion.getAlumno().getTerceraEdad().equals("N")
        &&
inscripcion.getEvento().getCurso().getNombreCurso()
        .equals("GIMNASIA PARA LA TERCERA EDAD"))
    {

```



```

        int                                cupodisEve                                =
inscripcion.getEvento().getCupoDisponible();
        inscripcion.getEvento().setCupoDisponible(cupodisEve - 1);
        double                            costoEve3Edad                            =
inscripcion.getEvento().getCostoEve();
        inscripcion.setValor(costoEve3Edad * 0.50);
        em.merge(inscripcion);
        em.merge(inscripcion.getEvento());
        cadenaInscripcion = "Matriculación ";
    } else if (inscripcion.getConfirmaIns() != null
        && inscripcion.getConfirmaIns().equals("S")
        && inscripcion.getAlumno().getCapacidadEspecial()
!= null
        &&
inscripcion.getAlumno().getCapacidadEspecial().equals("S")) {

        int                                cupodisEve                                =
inscripcion.getEvento().getCupoDisponible();
        double                            costoEve3Edad                            =
inscripcion.getEvento().getCostoEve();
        if (cupodisEve > 0) {
            inscripcion.getEvento().setCupoDisponible(cupodisEve
- 1);

            inscripcion.setValor(costoEve3Edad * 0.50);
            em.merge(inscripcion);
            em.merge(inscripcion.getEvento());
            cadenaInscripcion = "Matriculación ";
        }
    } else if (inscripcion.getConfirmaIns() != null
        && inscripcion.getConfirmaIns().equals("S")
        && inscripcion.getAlumno().getTerceraEdad() != null
        &&
inscripcion.getAlumno().getTerceraEdad().equals("S")) {

        int                                cupodisEve                                =
inscripcion.getEvento().getCupoDisponible();
        double                            costoEve3Edad                            =
inscripcion.getEvento().getCostoEve();
        if (cupodisEve > 0) {
            inscripcion.getEvento().setCupoDisponible(cupodisEve
- 1);

            inscripcion.setValor(costoEve3Edad * 0.50);
            em.merge(inscripcion);
            em.merge(inscripcion.getEvento());
            cadenaInscripcion = "Matriculación ";
        }
    } else if (inscripcion.getConfirmaIns() != null
        && inscripcion.getConfirmaIns().equals("N")
        && inscripcion.getAnulaIns() != null
        && inscripcion.getAnulaIns().equals("S")) {

```



```

        int cupodisEve =
inscripcion.getEvento().getCupoDisponible();
        if (cupodisEve > 0) {
            inscripcion.getEvento().setCupoDisponible(cupodisEve
+ 1);

            em.merge(inscripcion);
            em.merge(inscripcion.getEvento());
            cadenaInscripcion = "Anulación ";
        }
    } else if (inscripcion.getConfirmaIns() != null
        && inscripcion.getConfirmaIns().equals("S")) {

        int cupodisEve =
inscripcion.getEvento().getCupoDisponible();
        double costo = inscripcion.getEvento().getCostoEve();
        if (cupodisEve > 0) {
            inscripcion.getEvento().setCupoDisponible(cupodisEve
- 1);

            inscripcion.setValor(costo);
            em.merge(inscripcion);
            em.merge(inscripcion.getEvento());
            cadenaInscripcion = "Matriculación ";
        }
    } else if (inscripcion.getConfirmaIns() != null
        && inscripcion.getConfirmaIns().equals("N")
        && inscripcion.getAnulaIns() != null
        && inscripcion.getAnulaIns().equals("N")) {
        em.merge(inscripcion);
        em.merge(inscripcion.getEvento());
    } else {
        em.merge(inscripcion);
        em.merge(inscripcion.getEvento());
    }

    return cadenaInscripcion + " realizada correctamente";
}

@Override
public String eliminarInscripcion(Inscripcion inscripcion) throws Exception
{
    if (inscripcion.getConfirmaIns() != null
        && inscripcion.getConfirmaIns().equals("S")) {

        int cupodisEve =
inscripcion.getEvento().getCupoDisponible();
        if (cupodisEve > 0) {
            inscripcion.getEvento().setCupoDisponible(cupodisEve
- 1);

            em.merge(inscripcion);

```



```

        em.merge(inscripcion.getEvento());
    }
} else {
    em.persist(inscripcion);
    em.merge(inscripcion.getEvento());
}
return "Inscripción eliminada correctamente";
}

@SuppressWarnings("unchecked")
@Override
public List<Inscripcion> consultarInscripcionesAlumno(int idALum)
    throws Exception {
    Query conEveCur = em
        .createQuery("Select ins from Inscripcion ins where
ins.alumno.idALum =:idALum");
    conEveCur.setParameter("idALum", idALum);
    return conEveCur.getResultList();
}

@Override
public List<Object> consultarTodoEstado(String estado) {
    Query conEveCur = em
        .createQuery("Select eve from Evento eve where
eve.estadoEve =:estado");
    conEveCur.setParameter("estado", estado);
    return conEveCur.getResultList();
}

@SuppressWarnings("unchecked")
@Override
@ExcludeClassInterceptors
public List<Instructor> consultarInstructorOrdenPorNombre(String estado) {
    Query conInstructorOrd = em
        .createQuery("Select inst from Instructor inst where
inst.estado=:estado order by inst.apellidoInst asc");
    conInstructorOrd.setParameter("estado", estado);
    return conInstructorOrd.getResultList();
}

@SuppressWarnings("unchecked")
@Override
@ExcludeClassInterceptors
public List<Alumno> consultarAlumnoOrdenPorNombre(String estado) {
    Query conInstructorOrd = em
        .createQuery("Select alum from Alumno alum where
alum.estadoAlum=:estado order by alum.apellidoAlum asc");
    conInstructorOrd.setParameter("estado", estado);
    return conInstructorOrd.getResultList();
}

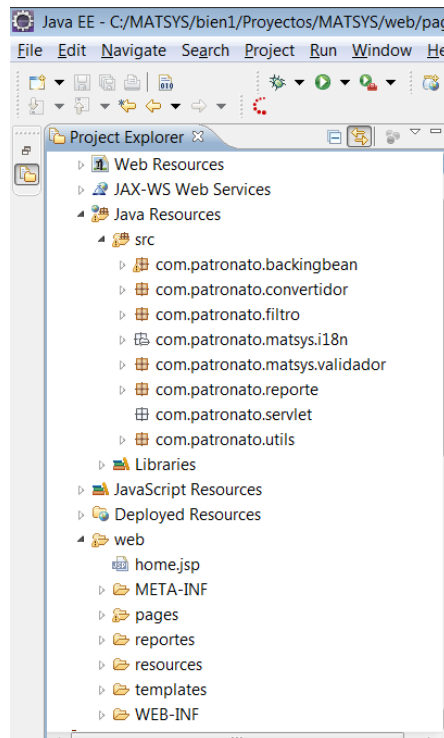
```



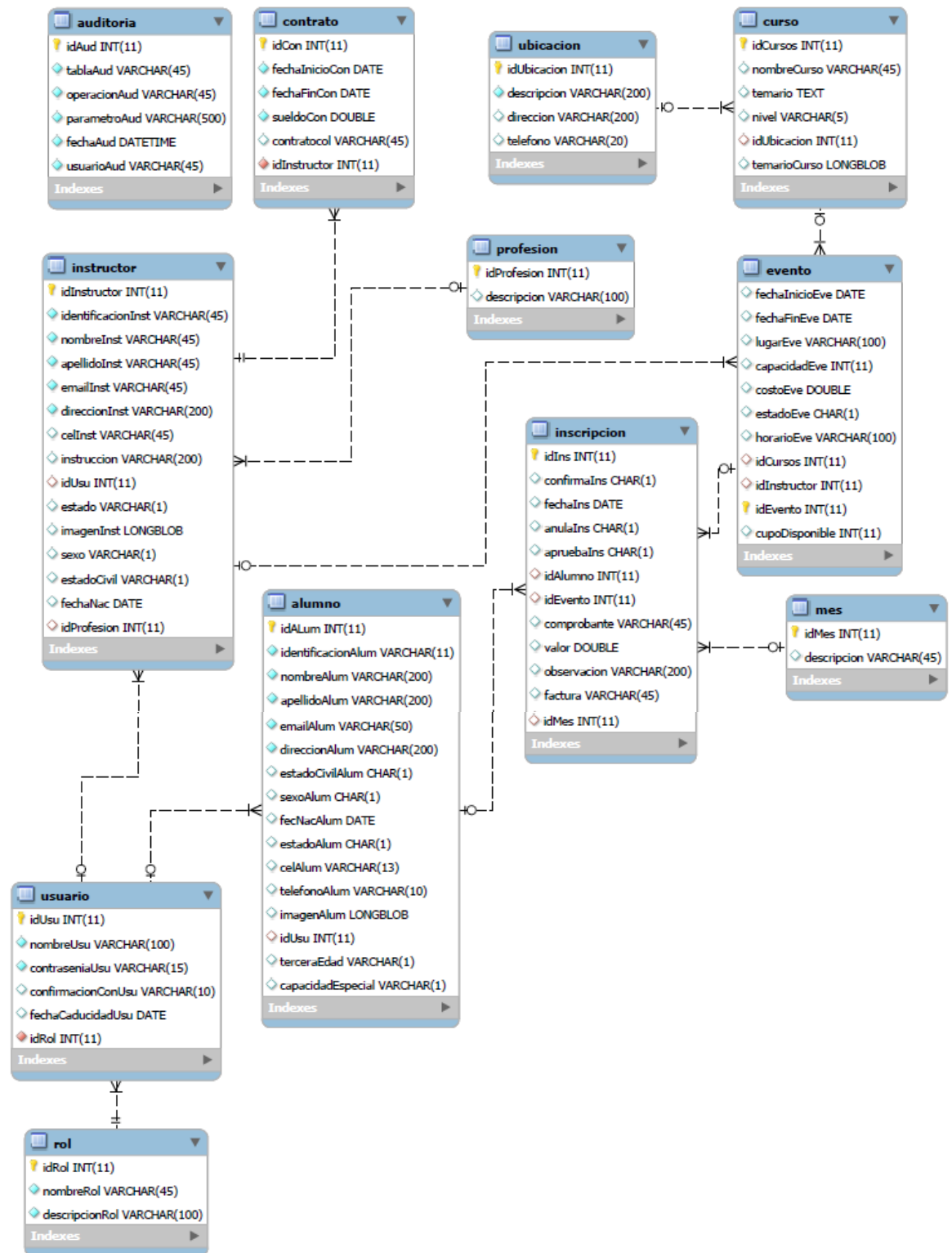
}

1.3 Capa de presentación

Define cuales son las clases que tienen que ser manejadas por el motor de JavaServer Faces, las clases que se van a utilizar para realizar conversiones, validaciones, filtro, reportes y la navegación entre páginas.



ANEXO C: BASE DE DATOS



ANEXO D: PRESUPUESTO

RUBROS DE GASTOS	CANTIDAD	VALOR (\$)	TOTAL
1 RECURSOS EMPRESARIALES			
PERSONAL DE APOYO: Coordinador del PPSRumiñahui	30h	---	---
SUBTOTAL		0.00	
2 RECURSOS HUMANOS			
Tutor de trabajo de graduación: Ing. César Morales Mejía Msc.	60h	---	---
Tribunal de trabajo de graduación	---	---	---
Tesista: Lorena Suntaxi	1000 h	1.00	1000.00
Encargado del área de sistemas (PPSRumiñahui): Ing. Romel Maroto	100 h	---	---
SUBTOTAL			1000.00
3 RECURSOS MATERIALES			
Material de escritorio:			
- Resma de papel	2	4.50	9.00
- Impresora Canon	1	120.00	120.00
- Cartucho de tinta B/N Canon	1	45.00	45.00
- Cartucho a color Canon	1	45.00	45.00
- PC de escritorio	1	800.00	800.00
Material bibliográfico:			
- Internet	6 meses	22.00	132.00
Transcripción borrador trabajo de grado	56h	1.00	56.00
Anillado plan de tesis	2	1.50	3.00
Empastado trabajo de grado	3	10.00	30.00
SUBTOTAL			1240.00
4 OTROS			
Transporte	--	50.00	50.00
Luz	--	20.00	20.00
SUBTOTAL			70.00
TOTAL GASTOS			2310.00
IMPREVISTOS (5%)			115.05
TOTAL DEL PRESUPUESTO			2425.05